IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 12, 2017, accepted July 10, 2017, date of publication July 17, 2017, date of current version October 25, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2727504

SparseLoc: Indoor Localization Using Sparse Representation

KONGYANG CHEN[1,2], YUE MI[1,3], YUN SHEN[1], YAN HONG[1,4], AI CHEN[1], AND MINGMING LU[5], (Member, IEEE)

[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
[2]Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen 518055, China
[3]China Mobile (Shenzhen) Company Ltd., Shenzhen 518048, China
[4]University of Science and Technology of China, Hefei 230026, China
[5]School of Information Science and Engineering, Central South University, Changsha 410083, China

Corresponding author: Mingming Lu (mingminglu@csu.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant 61379135, Grant 61232001, Grant 61173169, and Grant 60903222, in part by the China National Key Research and Development Program under Grant 2016YFB0502202, in part by the Major Science and Technology Research Program for Strategic Emerging Industry of Hunan under Grant 2012GK4054, in part by the National Science Foundation of Hunan under Grant 2016JJ2149, in part by the Shenzhen Scientific Research and Development Funds under Grant CXZZ20151117161747567, Grant KQCX20140520154115026, and Grant JCYJ20140610151856733, and in part by the SIAT Innovation Program for Excellent Young Researchers under Grant 201307.

ABSTRACT With the popularity of smart mobile devices, ‘‘context-aware’’ applications have attracted intense interest, for which location is one of the most essential contexts. Compared with outdoor localization, indoor localization has received much more attention from both academia and industry these days. Given the widespread use of WiFi hotspots, the received signal strength (RSS) fingerprint-based indoor localization technique is considered as a promising and practical solution because of its relatively high accuracy and low infrastructure cost. Inspired by our observation that sparsity is inherent to the WiFi signal, we present a new RSS fingerprint-based indoor localization approach, called *SparseLoc*. Through sparse representation of the fingerprints, SparseLoc can estimate a smart mobile device's location with a small error most of the time. Although the correlation between neighboring fingerprints affects the localization accuracy, SparseLoc uses the similarity between principal components of fingerprints to alleviate this effect. Based on the empirical experiments, we demonstrate that SparseLoc improves the localization accuracy by over 25% compared with the existing WiFi signal-based localization methods.

INDEX TERMS Indoor localization, RSS fingerprint, sparse representation, sparse dictionary, orthogonal matching pursuit.

I. INTRODUCTION

The widespread use of smart mobile devices and the increasing demand of context-ware services have attracted much attention on developing the localization techniques since the location information is a significant context in numerous applications [8], such as patient-care-aid in hospital, rescue-aid in disaster areas, and navigation for the tourists and mobile robots. Although the outdoor localization has been well established due to the commercial success of GPS, indoor localization is still in its early age. Numerous indoor localization techniques have been proposed. However, none of the existing techniques has been widely deployed.

Given the widespread use of the WiFi hotspots, the received signal strength (RSS) fingerprint based indoor localization technique is considered as a promising and practical solution because of its relatively high accuracy and low infrastructure cost. In the approach, a vector of RSSs from detectable Access Points (APs) is served as the fingerprint of a location. Then, the realtime RSS, called *online fingerprint*, is compared against the pre-stored fingerprint database to estimate the realtime location. Although existing RSS fingerprint based indoor localization technologies have obtained relatively high accuracy, it is still not sufficient to meet the high accuracy demand from some applications.

We have investigated the property of the RSS fingerprints and found strong *sparsity* for the RSS fingerprints at the same location. Through Principal Component Analysis (PCA) [10], the fingerprints at one location with 12 APs are transformed to 12 components (details in Section III). The magnitude of each component is shown in Figure 1. It can

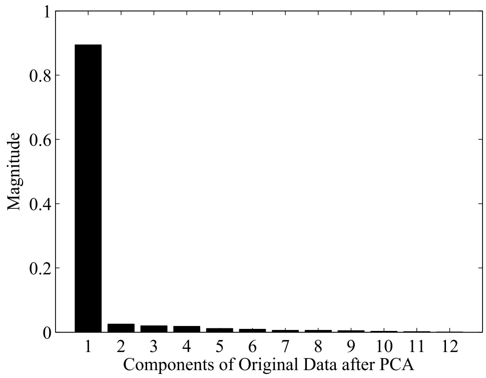


FIGURE 1. Magnitude of components of the fingerprints of one location through PCA.

been seen that one component has much bigger magnitude than others in the fingerprints. It means that the fingerprints at a location are compressible [3] and can be linearly represented by few basis vectors, i.e. they are *sparse*.

Motivated by *sparse representation* applied in face recognition [20], [23], we introduce sparse representation to indoor localization to leverage the sparsity in RSS fingerprints. In the image area, feature faces are extracted from the image database to sparsely represent the online images [24]. Faces that can be linearly represented by the current feature faces with a tiny residual would be classified to the current class. Because indoor localization can be considered as a classification problem, we may classify the fingerprints in a similar way.

A. PROBLEM STATEMENT

Motivated by the widespread availability of WiFi hotspots and smart mobile devices equipped with WiFi communication modula, we wish to implement a new WiFi based localization approach on the smart mobile devices that leverages the sparsity of RSS fingerprints to improve the indoor localization accuracy. To the best of our knowledge, this work is the first to utilize sparse representation in indoor localization.

However, because of the correlation between fingerprints of neighboring locations, directly using sparse representation leads to inaccurate positioning since the fingerprints of a location could sparsely represent the fingerprint of the neighboring location. This problem does not exist in face recognition since there is hardly any correlation between faces from different classes. Therefore, we must redesign the existing sparse representation algorithm to solve this problem.

In this paper, we propose a new indoor localization approach, named *SparseLoc*. SparseLoc is a RSS fingerprint based approach leveraging the sparsity of fingerprints. It first records the fingerprints of every reference point (location) and extracts a *sparse dictionary* for each point, which is used in sparse representation. For an online fingerprint, if it could not be approximately represented by a reference point's sparse dictionary, it would not be located to this reference point. In order to alleviate the effect of the correlation between fingerprints of neighboring locations, the mean fingerprint of a reference point is used in sparse representation. The principal component of the mean fingerprint is extracted by the sparse dictionary and compared to the principal component of the online fingerprint. An online fingerprint is located to a reference point only if the online fingerprint can be approximately represented by the sparse dictionary of this reference point and its principal component is similar to the principal component of the mean fingerprint of the reference point. Extensive experiments show that SparseLoc outperforms the traditional methods in localization accuracy by up to 25.7%.

The **main contributions** of this paper are as follows:

- We introduce *sparse representation* to indoor localization.
- We design the first indoor localization approach, called SparseLoc, that leverages the sparsity of RSS fingerprints.
- We implement SparseLoc on smart mobile devices.

B. ORGANIZATION

Section II introduces the preliminary of our approach. Section III analyzes the sparsity of RSS fingerprints. Some design considerations are elaborated in Section IV. Section V presents the design details of SparseLoc. In Section VI, we evaluate the performance of our approach through empirical experiments. Section VII presents some related work. Finally, Section VIII concludes the paper.

II. PRELIMINARY

This section introduces the concept of sparse representation and its application in face recognition, which is a typical application of sparse representation and has inspired us to do this work. This section also introduces an algorithm, called Orthogonal Matching Pursuit (OMP) [18], which is used in SparseLoc.

A. SPARSE REPRESENTATION

Sparse representation means that a high-dimensional signal is compressible and most information of this signal can be represented as a linear combination of a few elementary signals (bases) [7].

Consider a signal x in $\mathbb{R}^N$, it can be represented by a set of $N \times 1$ basis vectors (elementary signals) $\{\Psi_i\}_{i=1}^{N}$ as

$$\boldsymbol{x} = \boldsymbol{\Psi s} \tag{1}$$

In which, $\boldsymbol{s}$ is the coefficient vector. If some components of $\boldsymbol{s}$ are zero, the signal can be represented by parts of N elementary signals. The collection of the elementary signals with nonzero coefficients is called a sparse dictionary.

Sparse representation has been widely used in noise reduction, compression, feature extraction, pattern classification, etc. In pattern classification, a signal that can not be approximately represented by a class's dictionary is considered not belonging to this class.

B. APPLICATION IN FACE RECOGNITION

In face recognition, face images within the same category usually have similar features, and thusly they are highly correlated. The sparse-representation-based face recognition [20], [26] utilizes the above phenomenon to sparsely represent a high-dimensional face image $\boldsymbol{f}$ through a feature vector of much lower density called sparse dictionary $\boldsymbol{D}$. Formally, a sparse representation can be formalized as follows.

$$\boldsymbol{f} = \boldsymbol{Dc} + \boldsymbol{r} \tag{2}$$

where $\boldsymbol{c}$ denotes the vector sparse coefficients, and $\boldsymbol{r}$ represents the residual after sparse representation.

For a face image $\boldsymbol{f}_t$ to be classified, if it belongs to a class, it can be approximately represented by the sparse dictionary $\boldsymbol{D}$ of that class via a constrained l_1-minimization, as shown in Equation (3)

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \|\boldsymbol{c}\|_1 \quad s.t. \quad \|\boldsymbol{f}_t - \boldsymbol{Dc}\|_2 < \varepsilon \tag{3}$$

where ε represents the dense small noise in $\boldsymbol{f}_t$. If $\boldsymbol{f}_t$ belongs to a class, the residual $\boldsymbol{r} = \boldsymbol{f}_t - \boldsymbol{Dc}$ should be pretty small; otherwise, the corresponding residual would be relatively large. Thus, a face image can be classified based on the magnitude of the corresponding residual.

C. ORTHOGONAL MATCHING PURSUIT (OMP)

OMP is a greedy algorithm. Given a signal, a set of probable bases and the representation requirements, OMP is to find a subset of bases that can approximately represent the signal and meet the representation requirements.

For a signal with a set of probable bases, OMP first finds the basis nearest to the original signal as the first dictionary element. If there are already some bases in the dictionary, OMP first calculates the residual $\boldsymbol{r}$ of the signal after projecting the signal on the vector space expanded by the chosen bases in the dictionary as follows:

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}\in\mathbb{R}^k} \|\boldsymbol{f} - \boldsymbol{\Phi}\cdot\boldsymbol{c}\|_2 \tag{4}$$

$$\boldsymbol{r} = \boldsymbol{f} - \boldsymbol{\Phi}\cdot\hat{\boldsymbol{c}} \tag{5}$$

where $\boldsymbol{f}$ is the original signal and $\boldsymbol{\Phi}$ is the current dictionary containing the chosen bases. Then, among the unchosen probable bases, OMP chooses the basis nearest to $\boldsymbol{r}$ as the next dictionary element. The OMP algorithm does not stop until meeting all the representation requirements. The representation requirements can be the number of chosen bases reaches a certain value or the residual $\boldsymbol{r}$ is smaller than a certain value. Finally, OMP outputs the subset of the bases that have been chosen by the algorithm.

III. SPARSITY ANALYSIS

To bring in sparse representation, it should been shown that there exists *sparsity* in RSS fingerprints. Sparsity means that through some sort of transformation, most information of a high-dimensional signal exists in a limited number of dimensions.

We use Principal Component Analysis (PCA, also known as singular value decomposition [10]) to show the sparsity in RSS fingerprints. PCA uses orthogonal transformation to covert possibly correlated variables (signals) into a set of values of linearly uncorrelated variables called principal components. Given a collection of fingerprints $\boldsymbol{F} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_N]$, PCA finds the principal components in it. It first calculate the covariance matrix by

$$\boldsymbol{C} = \frac{\boldsymbol{F}^T\boldsymbol{F}}{N-1} \tag{6}$$

$\boldsymbol{C}$ is the covariance matrix. Then, PCA diagonalizes $\boldsymbol{C}$ to get

$$\boldsymbol{\Lambda} = \boldsymbol{P}^T\boldsymbol{CP} \tag{7}$$

where the diagonal elements in $\boldsymbol{\Lambda}$ mean the power of those components. The component with the biggest power is considered as the principal component.

We set an experiment on the 9th floor in our office building to analyze sparsity in fingerprints. We randomly choose a position as the reference point on the 9th floor. Over 100 fingerprints of the reference point are collected by an Android mobile phone. Each fingerprint consists of 12 RSS values of 12 APs. Through PCA, the fingerprints transformed to 12 components. The magnitude (ratio to the summation) of each component is shown in Figure 1. It can been seen that one component (principal component) has much bigger magnitude than others, and most of components are negligible. So, most information of these fingerprints exists in a limited number of dimensions after transformation and can be linearly represented by few basis vectors, i.e. they are *sparse*. We get the similar results for sparsity analysis at other positions.

IV. DESIGN RATIONALE

In this section, we show that the typical sparse representation method could not be directly applied to RSS fingerprint based indoor localization. We discuss the two challenges and provide the corresponding solutions.

A. CHALLENGES

The sparse-representation-based method described in Subsection II-B cannot be directly applied to fingerprint-based localization problem due the following two challenges: 1) the construction of the sparse dictionary; 2) the strong correlation of WiFi signal among nearby reference points.

In fingerprint-based indoor localization, given a collection of training samples ($\boldsymbol{F} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_N]$, in which $\boldsymbol{f}_i = [s_{i1}, \ldots, s_{iM}]$, s_{ij} is the RSS of the jth AP), it cannot be directly used as sparse dictionary, because the number of training samples N (collected through site survey) can be much larger than the dimension of RSS fingerprint M (the number of available APs). Thus, the collected training samples actually form a over-complete dictionary, which is not sparse. As a comparison, in face recognition, the number of collected training samples is usually much less than the dimension of face images.

Moreover, because of the WiFi propagation characteristic, fingerprints of nearby reference points may be highly correlated. We use Pearson product-moment correlation coefficient [12] to analyze the correlation between the fingerprints of nearby references points. Two positions on the 9th floor in our office building are chosen as the reference points. The distance between these two points is about 1 meter. Collections of 30 fingerprints corresponding to these two neighboring reference points are analyzed and the correlation coefficient is 0.7665, which means strong correlations between the fingerprints. We get the similar results for correlation analysis at other positions. Therefore, it is insufficient to distinguish RSS fingerprint through residual alone. As a comparison, in face recognition, face images are sparsely distributed in a high-dimensional space, which does not have such a high correlation phenomenon.

In the following subsections, we present our solutions to these two challenges.

B. TRAINING THE SPARSE DICTIONARY

From the above discussion, it is sufficient to find a subset of the collected training samples in order to form a sparse (incomplete) dictionary. The challenge is how to find an appropriate subset of the training samples. Before we discuss how to select the subset to construct the incomplete dictionary, we need to first give a formal definition of an incomplete dictionary, and discuss why incomplete dictionary is necessary.

1) INCOMPLETE DICTIONARY

Definition 1 (Incomplete Dictionary): Given a dictionary $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_k] \in \mathbb{R}^{M\times k}$, where any $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$ ($1 \le i, j \le k$) are linearly independent, if $k < M$, then $\boldsymbol{D}$ is incomplete.

Definition 1 can be described as: a dictionary $\boldsymbol{D}$ is an incomplete dictionary if the size of the dictionary (k) is smaller than the dimension of the fingerprints (M). As a comparison, if $k \geq M$, then $\boldsymbol{D}$ is over-complete.

The reason for our focus on incomplete dictionary is that, theoretically, every M-dimensional fingerprint can be completely represented by an over-complete dictionary, which results in zero residual. As the residual is the metric to do the classification task, the over-complete dictionary leaves no room for localization.

However, for a reference point, we can construct an incomplete dictionary $\boldsymbol{D}$ from the over-complete dictionary $\boldsymbol{X} = [x_1, \ldots, x_z]$ ($z \geqslant M$) consisting of all the collected training samples, which satisfies a l_2-minimization [26] as follows:

$$\min_{\boldsymbol{c}} \|\boldsymbol{f} - \sum_{i=1}^{z} x_i c_i\|_2 \quad s.t. \quad \|\boldsymbol{c}\|_0 = k \tag{8}$$

$$\boldsymbol{D} = \{x_j \mid \forall j, \quad c_j \neq 0\} \tag{9}$$

where $\boldsymbol{f}$ is another fingerprint (not in $\boldsymbol{X}$) of the given reference point, and l_0- norm $\|\boldsymbol{c}\|_0$ denotes the number of non-zero elements in the sparse coefficient vector. Those training fingerprints with non-zero coefficient represent the key features of the fingerprint $\boldsymbol{f}$. Therefore, they are chosen to from the sparse dictionary $\boldsymbol{D}$.

The basic idea to construct an incomplete dictionary is relatively simple. Any element in $\boldsymbol{X}$ with zero coefficient is removed. Thus, all elements in $\boldsymbol{X}$ with non-zero coefficients finally generate the incomplete dictionary.

Since directly applying Equation (8) to generate the incomplete dictionary is time-consuming, we employ OMP, a sub-optimal and greedy solution (the detailed description refers to Section II-C). OMP first finds the nearest fingerprint to $\boldsymbol{f}$ and selects it as the first dictionary element. Then, it repetitively projects $\boldsymbol{f}$ onto the current dictionary (Equation (4)), gets the residual $\boldsymbol{r}$ (Equation (5)), and chooses the nearest fingerprint to $\boldsymbol{r}$ as the next dictionary element until there are k elements in the dictionary.

2) SELECTION OF MAIN FINGERPRINT

According to Equation (8), the selection of the fingerprint $\boldsymbol{f}$ is also a key issue in training the sparse dictionary. $\boldsymbol{f}$ is called the *main fingerprint*. We can collect a fingerprint or choose the mean fingerprint (the mean value of all the training samples) as $\boldsymbol{f}$. We believe that the mean fingerprint might be the better choice. An arbitrary fingerprint may not be able to reflect the RSS characteristics of a given reference point due to the random noise. We believe the mean fingerprint can represent the characteristics more reliably since it has eliminated most white noise.

To verify the above argument, we conduct an experiment to show the superiority of the mean fingerprint. In this experiment, at each of the total 100 reference points, 15 fingerprints are collected as the training samples. For each reference point, we use the mean fingerprint of the 15 fingerprints and another collected fingerprint to train two sparse dictionaries, respectively. Then, each of the 15 fingerprints is sparsely represented by the two sparse dictionaries, respectively. Finally, the total 3000 residuals obtained through the sparse representations are used to calculate the mean and variance. The statistical magnitude of the residuals is used to evaluate the two sparse dictionaries. Smaller statistical magnitude of the residuals means that the corresponding dictionary is more representative.

The experiment results show that the mean magnitude of the residuals is 5.93 when using mean fingerprint, while the mean value of the residuals is 6.43 when using another collected fingerprint as the main fingerprint. Moreover, the variance of residuals is 23.8 when using mean fingerprint, which is much smaller than 33.05, the variance of residuals when using another collected fingerprint. This means the residuals is more stable when using mean fingerprint. As shown in Figure 2, 90% of the mean magnitude of the residuals is smaller than 12 when using mean fingerprint, while it is 13.7 when using another collected fingerprint as the main fingerprint. In summary, it is reasonable to use the mean fingerprint as the main fingerprint to generate a sparse dictionary.

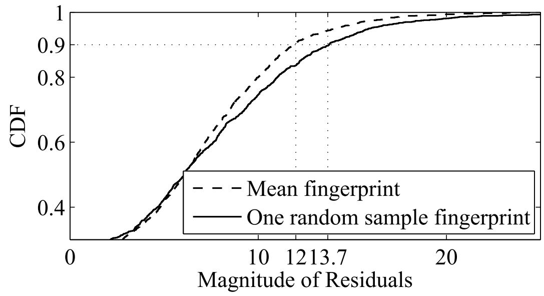


FIGURE 2. Cumulative distribution function of the Magnitude of the Residuals.

C. ADDRESSING THE CORRELATION ISSUE

As mentioned in Section IV-A, the propagation characteristic of WiFi signal results in a high correlation among the RSS fingerprints of nearby reference points, which reduces the significance of a small residual, and finally affects the localization accuracy.

When a fingerprint collected at a reference point A is sparsely represented by the sparse dictionary at the reference point B near to A according to Equation (3), the corresponding residual might also be relative small due to the correlation effect. Thus, the fingerprint collected at the reference point A might be considered as a fingerprint collected at the reference point B.

To mitigate the influence of the correlation, we introduce an additional metric, *principal component*, together with the residual, to determine the classification result. In this paper, the principal component $\boldsymbol{f}_\mu$ of a fingerprint $\boldsymbol{f}$ corresponding to a dictionary $\boldsymbol{D}$ is represented as

$$\boldsymbol{f}_{\boldsymbol{\mu}} = \boldsymbol{D} \cdot \hat{\boldsymbol{c}_\mu} \tag{10}$$

in which

$$\hat{\boldsymbol{c}_\mu} = \arg \min_{\boldsymbol{c}_\mu \in \mathbb{R}^k} \|\boldsymbol{f} - \boldsymbol{D} \cdot \boldsymbol{c}_\mu\|_2 \tag{11}$$

The rationale to choose the principal component as an additional metric is that the principal components of the two fingerprints associated with two nearby reference points are usually different from each other because the principal component reflects the key characteristic of a fingerprint.

To mitigate the correlation effect, the principal component of a fingerprint $\boldsymbol{f}_1$ is compared with the principal component of the fingerprint $\boldsymbol{f}_2$ collected at a reference point to determine whether $\boldsymbol{f}_1$ is collected at this reference point. Then, the issue is how to choose $\boldsymbol{f}_2$. Generally, $\boldsymbol{f}_2$ should reflect the general characteristic of the fingerprints collected at the given reference point. We can randomly pick a fingerprint or select the mean fingerprint. Based on the experiment results in subsection IV-B.2, we think the mean fingerprint is a preferable choice.

Hence, in our approach, once a mobile device needs to determine whether its location is close to a reference point, it first collects a RSS fingerprint $\boldsymbol{f}_t$, and projects the fingerprint $\boldsymbol{f}_t$ to the corresponding dictionary $\boldsymbol{D}$ of this reference point to get the coefficient vector $\hat{\boldsymbol{c}}_t$ by

$$\hat{\boldsymbol{c}_t} = \arg \min_{\boldsymbol{c}_\mu \in \mathbb{R}^k} \|\boldsymbol{f}_t - \boldsymbol{D} \cdot \boldsymbol{c}_t\|_2 \tag{12}$$

Then, the corresponding principal component $\boldsymbol{f}_{t\mu}$ and residual $\boldsymbol{r}$ can be calculated as

$$\boldsymbol{f}_{t\mu} = \boldsymbol{D} \cdot \hat{\boldsymbol{c}_t} \tag{13}$$

$$\boldsymbol{r} = \boldsymbol{f}_t - \boldsymbol{f}_{t\mu} \tag{14}$$

Similarly, the mean fingerprint of this reference point can be also projected to the dictionary $\boldsymbol{D}$ to get the principal component of the mean fingerprint, denoted as $\bar{\boldsymbol{f}}_\mu$. Thus, the principal component and residual can be combined to generate a joint metric l as

$$l = \lambda\|\boldsymbol{f}_{t\mu} - \bar{\boldsymbol{f}}_\mu\|_2 + (1-\lambda)\|\boldsymbol{r}\|_2 \tag{15}$$

where $\lambda \in (0, 1)$ is a tradeoff to balance the relative weights between the principal component and the residual. Generally speaking, the smaller the value of l is, the closer to the reference point the device is.

V. SPARSELOC

In this section, we present the detailed procedure of SparseLoc. First, Subsection V-A gives an overview of SparseLoc. Then, Subsections V-B and V-C present the detailed descriptions of the two phases of SparseLoc, i.e. training phase and matching phase, respectively.

A. OVERVIEW

As illustrated in Figure 3, SparseLoc has two phases: training phase and matching phase. In the training phase, SparseLoc trains a sparse dictionary at each reference point through OMP (the detailed description refers to Section II-C). At the end of the training phase, both the sparse dictionary and the principal component of the mean fingerprint at each reference point are recorded in the fingerprint database.

In the matching phase, a fingerprint to be localized is sparsely represented through the sparse dictionary at every reference point, and is divided into two components: the principal component and the residual. Through the joint metric l (defined in Equation (15)), the fingerprint's position is determined accordingly.

In the following subsections, Subsections V-B and V-C present the detailed descriptions of the training phase and matching phase, respectively.

B. TRAINING PHASE

In this phase, SparseLoc first selects some positions as the reference points. At every reference point, SparseLoc collects multiple fingerprints and records them into the fingerprint database. Based on the collected fingerprints, the mean fingerprint at each reference point can be calculated.

Then, at each reference point, SparseLoc uses the mean fingerprint to train the sparse dictionary. Formally, given a collection of fingerprints $\boldsymbol{F} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_N] \in \mathbb{R}^{M \times N}$ and

Algorithm 1 Training Phase of SparseLoc

Input:

N: number of fingerprints at each reference point;
p: number of reference points;
$\boldsymbol{F}_i = [\boldsymbol{f}_{i1}, \ldots, \boldsymbol{f}_{iN}] \in \mathbb{R}^{M\times N}, (i = 1, \ldots, p)$: the collections of training fingerprints of the i-th reference points.
k: the optimal size of the sparse dictionary.

Output:

$\boldsymbol{D}_i = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_k], (i = 1, \ldots, p)$: the Sparse dictionary;
$\boldsymbol{c}_i \in \mathbb{R}^k$: the coefficient vectors;
$\boldsymbol{f}_{i\mu}$: the principal component of the mean fingerprint at the i-th reference point;
$\boldsymbol{r}_i$: the residual of the mean fingerprint at the i-th reference point.

1: **for** the i-th reference point, $i = 1, \ldots, p$ **do**
2: Calculate the statistical mean value $\bar{\boldsymbol{f}}_i$ of fingerprints.

$$\bar{\boldsymbol{f}}_i = \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{f}_{i,j} \tag{16}$$

3: Initialize: $\boldsymbol{D}_i = \boldsymbol{0}^{M\times k}, \boldsymbol{c}_i = \boldsymbol{0}^k$,
set of fingerprint index:$S_i = \{1, 2, \ldots, N\}$.
4: Initialize residual: $\boldsymbol{r}_i = \bar{\boldsymbol{f}}_i$.
5: **for** $q = 1, 2, \ldots, k$ **do**
6: Find the index l of q-th dictionary element $\boldsymbol{d}_q$ from $\boldsymbol{F}_i$, which is nearest to current residual $\boldsymbol{r}_i$;

$$l = \arg\min_{j\in S_i} \|\boldsymbol{r}_i - \boldsymbol{f}_{ij}\| \tag{17}$$

$$\begin{gathered}\boldsymbol{d}_{ip} = \boldsymbol{f}_{il}\\ S_i = S_i - \{l\}\end{gathered} \tag{18}$$

7: Get the coefficients $\boldsymbol{c}_i$ and the residual $\boldsymbol{r}_i$ of representation of the current dictionary by projecting the mean fingerprint $\bar{\boldsymbol{f}}_i$ to the current dictionary $\boldsymbol{D}_i$ with the least square method:

$$\boldsymbol{c}_i = \arg\min_{\boldsymbol{c}\in\mathbb{R}^k} \|\bar{\boldsymbol{f}}_i - \boldsymbol{D}_i\cdot\boldsymbol{c}\|_2 \tag{19}$$

$$\boldsymbol{r}_i = \bar{\boldsymbol{f}}_i - \boldsymbol{D}_i\cdot\boldsymbol{c}_i \tag{20}$$

8: **end for**
9: Get the residual $\boldsymbol{r}_i$ and the principal component of the mean fingerprint $\boldsymbol{f}_\mu$ of the i-th reference point.

$$\begin{gathered}\boldsymbol{r}_i = \bar{\boldsymbol{f}}_i - \boldsymbol{D}_i\cdot\boldsymbol{c}_i\\ \boldsymbol{f}_{i\mu} = \boldsymbol{D}_i\cdot\boldsymbol{c}_i\end{gathered} \tag{21}$$

10: **end for**

the mean fingerprint $\bar{\boldsymbol{f}}$, the sparse dictionary can be obtained through the optimal algorithm described in Equation (8) or the sub-optimal algorithm called OMP. Since the optimal solution of Equation (8) is time-consuming, SparseLoc uses OMP (refer to Section II-C) to balance between solution quality and time efficiency.

Once the sparse dictionary $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_k] \in \mathbb{R}^{M\times k}$ has been determined, SparseLoc will sparsely represent the mean fingerprint according to Equation (2), i.e., $\bar{\boldsymbol{f}} = \boldsymbol{Dc} + \boldsymbol{r}$, where $\bar{\boldsymbol{f}}$ denotes the mean fingerprint. Furthermore, through the l_1-minimization formalized in Equation (3), a mean fingerprint is partitioned into two parts: 1) principal component $\bar{\boldsymbol{f}}_\mu \in \mathbb{R}^M$, where $\bar{\boldsymbol{f}}_\mu = \boldsymbol{Dc}$, and 2) residual $\boldsymbol{r} \in \mathbb{R}^M$, where $\boldsymbol{r} = \bar{\boldsymbol{f}} - \bar{\boldsymbol{f}}_\mu$. Because the residual of mean fingerprints is mainly caused by the white noise, SparseLoc only records the principal components.

Since Equation (2) does not present an algorithm to obtain a principal component, we need to find a way to compute $\bar{\boldsymbol{f}}_\mu$. Because, for a given reference point, its sparse dictionary is fixed, to find $\bar{\boldsymbol{f}}_\mu$ is equal to determine the appropriate vector of sparse coefficients $\boldsymbol{c}$.

Once the sparse dictionary has been established, the vector of sparse coefficients that represents $\boldsymbol{f}$ can be calculated according to Equation (22). Here, $\|\boldsymbol{c}\|_1$ is l_1-norm, which means a sample could faithfully be represented with least elements. $\|\boldsymbol{f} - \boldsymbol{Dc}\|_2^2$ means the fingerprint is represented by the sparse dictionary with a small residual. Since the l_1-minimization process is time-consuming, a more efficient l_2-minimization process as Equation (23) has been proposed, which uses collaborative representation [26]. Here, the l_2-norm leads to almost the same result as l_1-norm. Thus, we adopt the regularized least square method inherent to Equation (23) to get the optimal $\boldsymbol{c}$, which in turn can calculate the appropriate principal component.

$$\hat{\boldsymbol{c}} = \arg\min_{c}\{\|\boldsymbol{f} - \boldsymbol{Dc}\|_2^2 + \sigma\|\boldsymbol{c}\|_1\} \tag{22}$$

$$\hat{\boldsymbol{c}} = \arg\min_{c}\{\|\boldsymbol{f} - \boldsymbol{Dc}\|_2^2 + \sigma\|\boldsymbol{c}\|_2^2\} \tag{23}$$

In summary, once the sparse dictionary is identified, SparseLoc sparsely represents the mean fingerprint $\bar{\boldsymbol{f}}$ through the regularized least square method, and obtain the principal component of the mean fingerprint accordingly.

At the end of the training phase, both of the sparse dictionary $\boldsymbol{D} = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_k]$ and the principal component of the mean fingerprint $\bar{\boldsymbol{f}}_\mu$ will be recorded in the fingerprint database.

The algorithm of training phase in SparseLoc is illustrated as Algorithm 1. In Step 2, SparseLoc first calculates the mean fingerprint of the reference point. Then, in Steps 4-8, the algorithm uses the mean fingerprint to train an incomplete dictionary from the training fingerprints. To train the dictionary, the algorithm employs the OMP [18] to find the sub-optimal solution. Once the sparse dictionary has been established, SparseLoc projects the mean fingerprint onto the dictionary in Step 7. Finally, the residual and the principal component are calculated by the dictionary in Step 9. Both of the sparse dictionary and the principal component of the mean fingerprint are recorded in the fingerprint database.

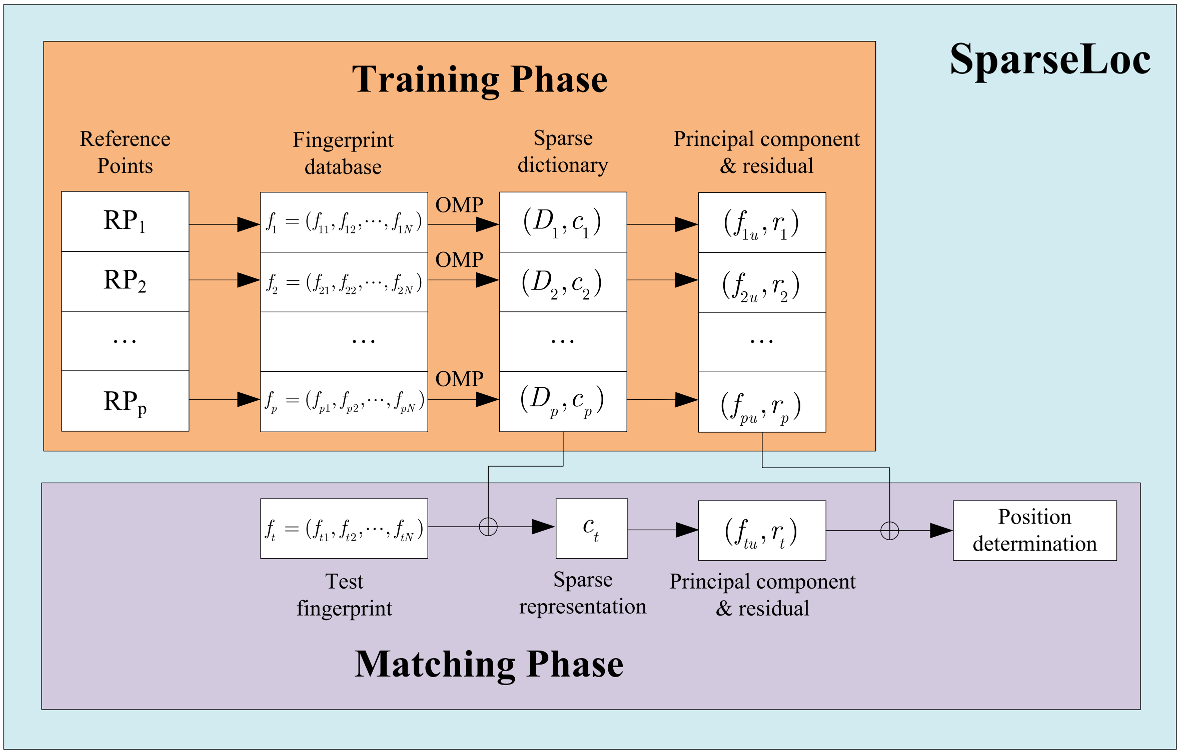


FIGURE 3. **Overview of SparseLoc.**

Computational Complexity: The computational complexity of Equation (16) is $O(MN)$, where M is the dimension of fingerprints and N is the number of fingerprints of a reference point. Equation (17) has the same computational complexity. The complexity of Equation (19) is $O(Mk^2)$, where k is the Sparse dictionary size. In a summary, the algorithm complexity is $O(pMk^3 + pkMN)$, in which p is the number of reference points. Moreover, the training phase can run on a server, which significantly reduces the overhead of mobile devices.

C. MATCHING PHASE

In fingerprint-based localization, the "logical" distance between fingerprints represents the physical distances between the corresponding localization points. So, the distance formula to calculate the "logical" distances between fingerprints is crucial for the localization accuracy.

In SparseLoc, a fingerprint is sparsely represented through a sparse dictionary, and the sparse representation partitions the fingerprint into two parts: the principal component and residual. Given a fingerprint $\boldsymbol{f}_t$ to be localized, SparseLoc first projects it to the incomplete dictionary $\boldsymbol{D}$ of each reference point in the form of $\boldsymbol{f}_t = \boldsymbol{D}\boldsymbol{c}_t + \boldsymbol{r}_t$, where $\boldsymbol{f}_{t\mu} = \boldsymbol{D}\boldsymbol{c}_t$ is the principal component, and $\boldsymbol{r}_t$ is the corresponding residual.

As analyzed in Section IV-C, a fingerprint $\boldsymbol{f}_t$ being classified to a location must meet two requirements: 1) the residual should be small, which means the fingerprint could be approximately represented by the corresponding dictionary; 2) the principal components of the mean fingerprint and $\boldsymbol{f}_t$ should be similar, so that the correlation among nearby reference points would not affect the localization accuracy.

The algorithm of matching phase of SparseLoc is illustrated as Algorithm 2. For each reference point, SparseLoc calculates its logical distance to $\boldsymbol{f}_t$. After loading the sparse dictionary from the fingerprint database, SparseLoc projects the $\boldsymbol{f}_t$ onto the dictionary in Step 2. Then, in Step 3, SparseLoc obtains the residual and the corresponding principal component. In Step 4, the algorithm uses a new metric to calculate the logical distance.

After SparseLoc obtains the distance between the online fingerprint and each reference point, it selects the reference point with the smallest logical distance as the target location.

Computational Complexity: The computational complexity of Equations (24), (25), and (26) are $O(Mk^2)$, $O(Mk)$, and $O(M)$, respectively. Hence, the computational complexity of Algorithm 2 is $O(Mpk^2)$. This part of algorithm runs on the mobile device.

VI. EXPERIMENTS AND EVALUATION

This section describes the details on experimental evaluation of SparseLoc. Real RSS data were collected in an office building. The performance is evaluated from many different aspects. The localization error is defined as the Euclidean distance between the estimated location and the corresponding actual location. We compare SparseLoc with two well known indoor localization approaches, RADAR [2]

Algorithm 2 Matching Phase of SparseLoc

Input:

p: number of reference points;
$\boldsymbol{f}_t$: the online fingerprint;
$\boldsymbol{D}_i$: the Sparse dictionary of the i-th reference point;
$\boldsymbol{f}_{i\mu}$: the principal component of the mean fingerprint at the i-th reference point;
$\boldsymbol{r}_i$: the residual of the mean fingerprint at the i-th reference point.

Output:

l_i: the logical distance from the online fingerprint to the i-th reference point.

for the i-th reference point, $i = 1, \ldots, p$ **do**

Project the online fingerprint $\boldsymbol{f}_t$ to the dictionary $\boldsymbol{D}_i$ with the least square method.

$$\boldsymbol{c}_t = \arg\min_{\boldsymbol{c}\in\mathbb{R}^k} \|\boldsymbol{f}_t - \boldsymbol{D}_i \cdot \boldsymbol{c}\|_2 \tag{24}$$

Get the residual $\boldsymbol{r}_t$ and principal component $\boldsymbol{f}_{t\mu}$ at the i-th reference point.

$$\begin{aligned}\boldsymbol{r}_t &= \boldsymbol{f}_t - \boldsymbol{D}_i \cdot \boldsymbol{c}_t\\ \boldsymbol{f}_{t\mu} &= \boldsymbol{D}_i \cdot \boldsymbol{c}_t\end{aligned} \tag{25}$$

Calculate the distance l_i

$$l_i = \lambda\|\boldsymbol{f}_{i\mu} - \boldsymbol{f}_{t\mu}\|_2 + (1-\lambda)\|\boldsymbol{r}_t\|_2 \tag{26}$$

end for

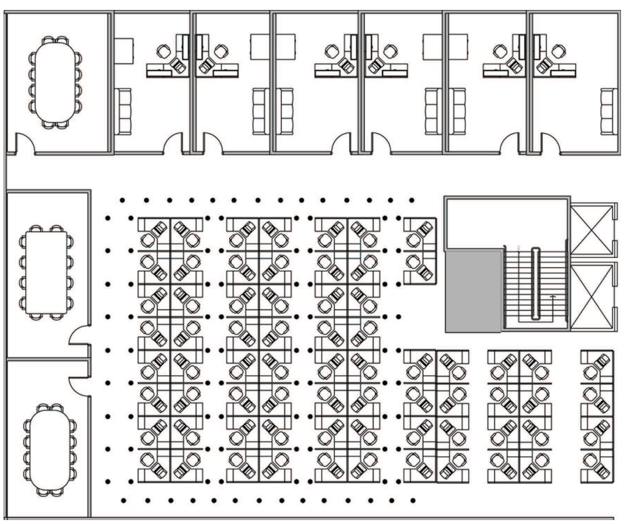


FIGURE 4. The Testbed.

(a RSS fingerprint based approach) and EZ [6] (a RSS model based approach).

A. EXPERIMENTS SETUP

We built the testbed on the 9th floor in an office building, illustrated in Figure 4. The experiments were carried out in an open space with the area of $272m^2$. 12 APs were visible in the area. For our experiments, we used a mobile phone (Motorola ME722 with Android 2.3) to collect the fingerprints. The software for fingerprint collection was developed with the Android application program interface. 100 reference points were labeled in the area. 15 fingerprints were collected at each reference point to form the fingerprint database and 5 online fingerprints at each of 80 locations were used to evaluate the localization accuracy, unless stated otherwise.

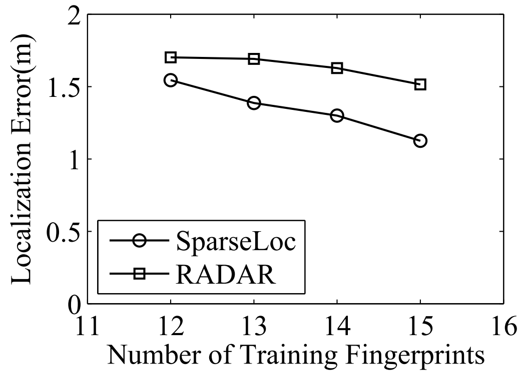


FIGURE 5. Performance of SparseLoc and RADAR on different fingerprint database size.

B. TRAINING DATABASE

In this subsection, we designed experiments to evaluate the influence of the size of training database.

1) TRAINING EXPERIMENT 1

In this experiment, We varied the number of training fingerprints collected at each reference point from 12 to 15 and used them to train the sparse dictionary. Other 5 fingerprints at 100 locations are collected as the testing online fingerprints to evaluated the performance of the sparse dictionaries. Figure 5 shows the mean localization error of RADAR and SparseLoc in the tests. Both of the algorithms perform stably in their mean localization error. And SparseLoc also shows a significant improvement tendency when increasing the number of training samples.

2) TRAINING EXPERIMENT 2

In this experiment, 25%, 50%, 75%, or 100% of the reference points were randomly chosen and 5 more fingerprints were collected at these points to train the optimal dictionary size k and λ. Figure 6 illustrates the mean localization errors with different fractions of training points. The results are stable enough, which infers that SparseLoc could use only part of the reference points to train the parameters k and λ. So, we decided to use only 25% of the points to train the parameters of SparseLoc in order to reduce the preliminary work. And we got $k = 6$ with $\lambda = 0.42$, which were used in the following experiments.

C. DISTANCE METRIC

In this experiment we compared the performances of four distance metrics. The first metric is the metric of SparseLoc. The second metric uses coefficients as the only input as $l = \|\boldsymbol{Dc} - \boldsymbol{Dc}_t\|_2$, while the third metric uses the residuals

TABLE 1. Mean localization errors for different metrics.

Area	SparseLoc	Coefficient Only	Residual Only	SR
Corridor 1	$1.977m$	$2.100m$	$2.509m$	$2.302m$
Corridor 2	$0.541m$	$0.844m$	$1.885m$	$1.336m$
Corridor 3	$0.775m$	$1.638m$	$2.428m$	$1.747m$
Corridor 4	$1.293m$	$2.111m$	$2.900m$	$2.007m$

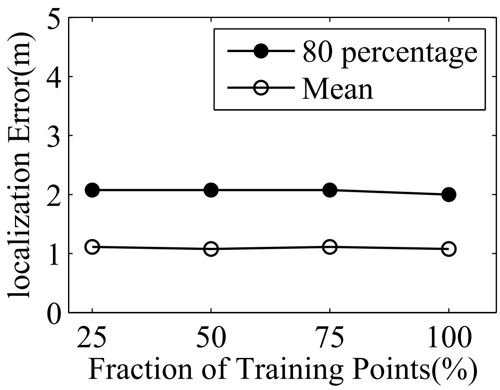


FIGURE 6. Performance of SparseLoc on Different Size of Training Points. The result remained stable on different sizes.

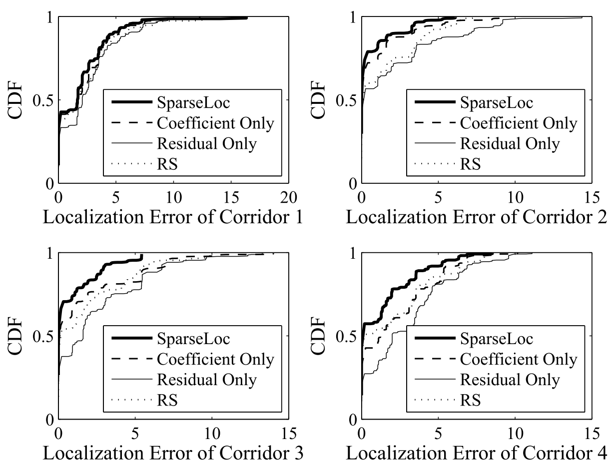


FIGURE 7. Cumulative Distribution Functions of Localization Errors for Different Distance Metrics.

as $l = \|\boldsymbol{r}\|_2$. And the last one is the metric of sparse representation (SR) according to Equation (22) as $l = \lambda\|\boldsymbol{c}_t\|_2 + (1-\lambda)\|\boldsymbol{r}_t\|_2$. The experiment was set on four different corridors as shown in Figure 4. The mean localization errors are shown in Table 1. And Figure 7 depicts the cumulative distribution function of localization errors for each metric. The metric of SparseLoc has better performance compared to other metrics.

D. ALGORITHM EVALUATION

The localization accuracy of SparseLoc is evaluated in this subsection. It is compared with two RSS-based indoor localization algorithms: RADAR and EZ.

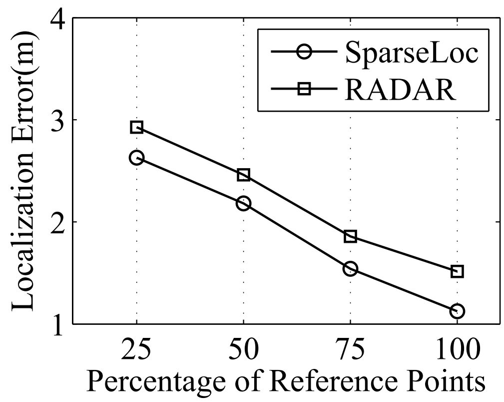


FIGURE 8. Localization Errors of Different Reference Point Densities.

1) LOCALIZATION EXPERIMENT 1

We first did an experiment by varying the density of the reference points. In the experiment, we picked up 25%, 50%, 75% or 100% reference points from the original fingerprint database by expanding the spacing between the points. The corresponding mean localization errors are showed in Figure 8. As expected, the errors decrease when the number of reference points increase. SparseLoc always performs better than RADAR in different densities of reference points.

2) LOCALIZATION EXPERIMENT 2

In this experiment, the mean values of the localization errors of SparseLoc, RADAR and EZ are evaluated. The mean localization error of SparseLoc is $1.125m$, 25.7% smaller than $1.515m$ of RADAR, while EZ presents a mean localization error of 8.06m. Note that EZ takes a long time to adaptively optimize its results. So, its performance is not good in our short period experiments. In addition, the standard deviation of SparseLoc is $1.91m$ while the standard deviation of RADAR is $2.33m$.

Then, the Cumulative Distribution Function (CDF) of localization errors for each algorithm is drawn in Figure 9. It can be concluded that about 80% fingerprint localization errors are smaller than $2m$ for SparseLoc compared to 70% for RADAR. And 90% of the errors for SparseLoc is smaller than $3.5m$ and RADAR reaches 90% at more than $5m$. In addition, EZ shows a clearly worst performance in our environment according to Figure 9.

In addition, a detailed error histogram is shown in Figure 10. It could be concluded that the concrete quantity

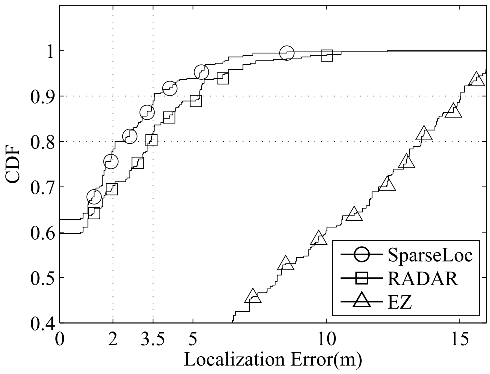


FIGURE 9. The Cumulative Distribution Function of Localization Errors of the Online Testing Fingerprints. 90% of the errors in SparseLoc is smaller than 3.5m and RADAR reaches 90% at more than 5m.

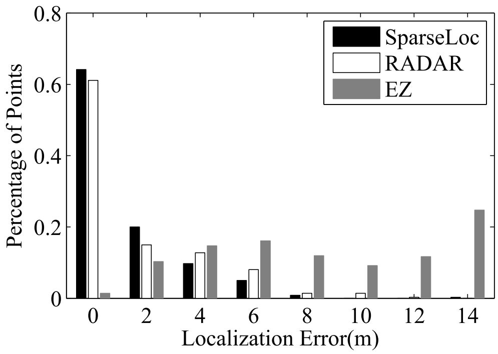


FIGURE 10. The Histogram of Errors of the Online Testing Fingerprints. The quantity of errors smaller than 3m for SparseLoc is larger than the quantities for RADAR and EZ.

of errors smaller than 3m for SparseLoc is larger than the quantity for RADAR, while EZ shows no superiority in either small errors or large errors. The quantity of errors larger than 3m for RADAR always exceed those for SparseLoc. Thus, compared to RADAR and EZ, SparseLoc performs better in localization accuracy.

Then, the localization errors of each online fingerprint is compared among different approaches. Because EZ shows a relatively worst performance, we only compare the localization errors of RADAR and SparseLoc. In this experiment, a value $ratio = (E_{RADAR} - E_{SparseLoc})/E_{SparseLoc}$ is calculated for each online fingerprint, where E_{RADAR} and $E_{SparseLoc}$ denote the localization errors for RADAR and SparseLoc, respectively. The CDF of the $ratio$ is shown in Figure 11.

Figure 11 shows that in less than 8% cases the localization errors for RADAR are smaller than the ones for SparseLoc, with $ratio \leq 0$. However, more than 20% cases the localization errors for RADAR are more than 50% larger than the ones for SparseLoc, with $ratio \geq 0.5$.

With the experiment results above, SparseLoc performs better than RADAR and EZ in many aspects.

VII. RELATED WORK

These days, indoor localization has become a very active research area. In this section, we present a brief overview of the existing works in this area.

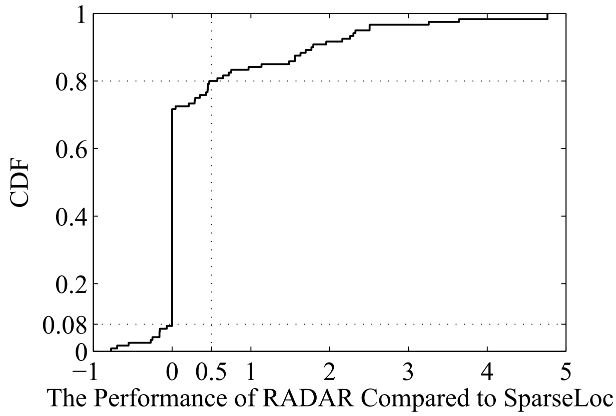


FIGURE 11. The Performance of RADAR Compared to SparseLoc at Each Reference Point. 20% of the localization errors for RADAR are more than 50% larger than the ones for SparseLoc.

A. SPECIALIZED INFRASTRUCTURES OR EQUIPMENTS

Some technologies rely on specialized infrastructure or equipments. For example, an ultrasound based system named Cricket [16] and an RFID based system named LANDMARC [15] require additional hardware infrastructures being deployed at various locations as well as on the localization devices. FM-based technology [5] and CSI-based technology [22] make use of existing signals in indoor environments. Visible light based positioning systems [4], [13], [14] are also presented in recent years. However, they need additional equipments deployed on the localization devices to collect the signals.

B. INERTIAL SENSING

Inertial sensing does not rely on additional infrastructures. It uses the sensors such as accelerometer, gyroscope and barometer to sense human motion so as to estimate the location. However, it has been proved that the localization error of inertial sensing accumulates. Therefore, some approaches to prevent the accumulation of errors were proposed in [1], [17], and [19]. A symbolic algorithm, called WILL [21], uses inertial sensing to draw the topology of the tracks of devices, and matches it with the topology of the environment. However, inertial sensing has to get the initial location of the object to do localization.

C. RSS-BASED INDOOR LOCALIZATION

RSS-based technologies use the RSS to locate the target device. Specifically, there are two main approaches of the technology: model-based and fingerprint-based.

1) MODEL-BASED APPROACHES

In model-based approaches, there are many geometrical models that reflect the signal attenuation caused by the propagation distance such as TIX [9]. EZ [6] is a model-based approach, which could train a model automatically. However, these model-based approaches typically earned poor accuracy.

2) FINGERPRINT-BASED APPROACHES

Fingerprint-based approaches record the RSS vectors as the fingerprints of reference locations and compare the realtime RSS against the pre-stored fingerprint database to estimate the realtime location [11], [25]. One simple and practical solution of fingerprint-based approach is RADAR [2]. Fingerprint-based approaches are also fused with inertial sensors [27] to provide better localization accuracy. For its relatively high accuracy and low hardware infrastructure requirement, fingerprint-based indoor localization technology has been accepted as an effective method in indoor localization. However, some applications require higher accuracy, which is hard to be satisfied by the existing approaches.

In this paper, we introduce a new indoor localization technology using sparse representation, which takes advantage of the sparsity in fingerprints. Compared to existing works, SparseLoc can get higher accuracy by using the widespread WiFi infrastructure.

VIII. CONCLUSION

SparseLoc is a novel RSS fingerprint based indoor localization approach, which is the first one to apply sparse representation in indoor localization. It does not require any specialized infrastructures or equipments. Because of the correlations between the RSS fingerprints at neighboring locations, we have introduced a new metric and a new algorithm to make sparse representation workable for indoor localization.

To evaluate the performance of SparseLoc, we have implemented the approach on mobile devices. Fingerprints were collected from real WLAN environments. We evaluated the performance of SparseLoc and compared the approach with RADAR and EZ. The average localization accuracy is improved by more than 25%. In addition, the computation complexity of SparseLoc is feasible to implement the approach on most of the off-the-shelf smart mobile devices.

REFERENCES

[1] H. Abdelnasser *et al.*, "SemanticSLAM: Using environment landmarks for unsupervised indoor localization," *IEEE Trans. Mobile Comput.*, vol. 15, no. 7, pp. 1770–1782, Jul. 2016.

[2] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. INFOCOM*, vol. 2. Mar. 2000, pp. 775–784.

[3] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.

[4] B. Xie *et al.*, "LIPS: A light intensity-based positioning system for indoor environments," *ACM Trans. Sensor Netw.*, vol. 12, no. 4, p. 28, 2016.

[5] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services*, 2012, pp. 169–182.

[6] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 173–184.

[7] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3320–3325, Dec. 2003.

[8] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 13–32, 1st Quart., 2009.

[9] Y. Gwon and R. Jain, "Error characteristics and calibration-free techniques for wireless LAN-based location estimation," in *Proc. 2nd Int. Workshop Mobility Manage. Wireless Access Protocols*, 2004, pp. 2–9.

[10] I. Jolliffe, *Principal Components Analysis*. New York, NY, USA: Wiley, 2005.

[11] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," in *Proc. INFOCOM*, vol. 2. Mar. 2004, pp. 1012–1022.

[12] M. G. Kendall, "Rank and product-moment correlation," *Biometrika*, vol. 36, pp. 177–193, Jun. 1949.

[13] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2014, pp. 447–458.

[14] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A visible light based positioning system," in *Proc. 11th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2014, pp. 331–343.

[15] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Netw.*, vol. 10, no. 6, pp. 701–710, 2004.

[16] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, 2000, pp. 32–43.

[17] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 293–304.

[18] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[19] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services*, 2012, pp. 197–210.

[20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[21] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: Wireless indoor localization without site survey," in *Proc. INFOCOM*, Mar. 2012, pp. 64–72.

[22] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. M. Ni, "FILA: Fine-grained indoor localization," in *Proc. INFOCOM*, Mar. 2012, pp. 2210–2218.

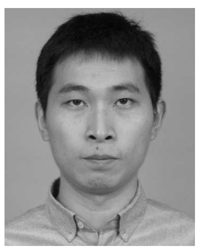
[23] J. Xie, L. Zhang, J. You, and D. Zhang, "Texture classification via patch-based sparse texton learning," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 2737–2740.

[24] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[25] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. 3rd Int. Conf. Mobile Syst., Appl., Services*, 2005, pp. 205–218.

[26] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 471–478.

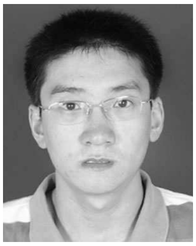
[27] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao, "Travi-Navi: Self-deployable indoor navigation system," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2014, pp. 471–482.



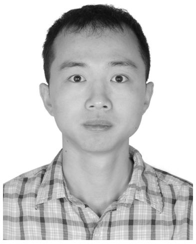
KONGYANG CHEN received the B.S. and M.E. degrees from Central South University, China. He is currently pursuing the Ph.D. degree with the Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, China. His research interests include wireless sensor network and mobile sensing.

YUE MI received the M.E. degree from the University of Chinese Academy of Sciences, China, in 2017. She is currently an Engineer with China Mobile (Shenzhen) Ltd., China. Her research interest is wireless communication.

YUN SHEN received the M.E. degree from the University of Chinese Academy of Sciences, China, in 2014. She is currently an Engineer with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China. Her research interest is indoor localization.



YAN HONG received the M.E. degree from the University of Science and Technology of China in 2013. He is currently an Engineer with Baidu Inc., China. He works in the area of mobile computing and wireless network.



AI CHEN received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer science and engineering from The Ohio State University, USA. He is currently an Associate Researcher with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Science. His research interests include wireless and sensor networks, mobile computing, and smartphone applications.

MINGMING LU (M'15) received the Ph.D. degree in computer science from Florida Atlantic University, USA, in 2008. He is currently an Associate Professor with Central South University. He has published over 40 scholarly journal and conference papers. His research areas include big data, Internet of Things, opportunistic sensing, compressive sensing, wireless networking, network coding, and localization. He was a Technical Program Committee Member of the Institute of Electrical and Electronics Engineers INFOCOM 2011 and one of the reviewers of numerous international journals and conferences, such as the IEEE Transactions on Parallel Distributed Systems, the IEEE Transactions on Mobile Computing, International Conference on Network Protocols, and MobiHoc.