Computer Communications 000 (2015) 1–8

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Reducing fingerprint collection for indoor localization☆

Zhuan Gu [a], Zeqin Chen [a], Yuexing Zhang [a], Ying Zhu [a], MingMing Lu [b], Ai Chen [a],*

[a] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, PR China
[b] Central South University, Changsha 410083, PR China

ARTICLE INFO

Article history:
Received 8 February 2015
Revised 27 August 2015
Accepted 21 September 2015
Available online xxx

Keywords:
Compressive sensing
Fingerprint collection
Indoor localization
Interpolation
Merging matrix

ABSTRACT

A typical WiFi-based indoor localization technique estimates a device's location by comparing received signal strength indicator (RSSI) against stored fingerprints and finding the closest matches. However, the collection of fingerprints is notoriously laborious and costly. It is challenging to reduce fingerprint collection and recover missing data without introducing significant errors. In this article, a novel approach based on compressive sensing is presented for recovering absent fingerprints. The hidden structure and redundancy characteristics of fingerprints are revealed in a merging matrix. The spatial and temporal correlations of fingerprints result in a small rank of the merging matrix. The *Sparsity Rank Singular Value Decomposition* (SRSVD) method is used to effectively reduce the interference caused by the multipath effect of the WiFi signal. We further propose to combine SRSVD with the *K*-Nearest Neighbor (KNN) algorithm to deal with missing columns or rows in the matrix. Experimental results show that with only half of the fingerprints, our approach can recover all the fingerprint information with error rate below 6.6%. Even with only 5% of the data, the approach can recover the information with error rate below 14%, without loss of localization accuracy.

1. Introduction

Numerous indoor localization systems are based on wireless local area networks (WLANs), which are ubiquitously deployed in public places [2]. The localization techniques underlying these systems can be generally classified into two categories: deterministic [3–5] and probabilistic [6,7]. In these systems, one has to measure received signal strength indicator (RSSI) values from surrounding access points (APs) at each reference location to construct a fingerprint database, which is a tedious and time-consuming process. For example, in our experiments, it takes 10 h for us to collect the fingerprint data of an office area of 1000 m^2. This problem seriously affects the application of indoor localization systems. In order to reduce the cost, several approaches [8,9] have been proposed. However, these approaches only attempt to reduce the number of reference points at which fingerprints are collected. At each point, the same amount of time still needs to be spent on collecting a stable RSSI result as in the naive approach.

In this article, we focus on reducing fingerprint measurements in both the time and space domains, and recovering the absent data faithfully. The main challenge is to keep a small localization error while recovering the absent data. Intuitively, we can consider the simple interpolation approach, such as the *K*-Nearest Neighbors (KNN) method. However, the relation between an absent data point and its neighbors is not easy to identify. Moreover, if a large number of data points are missed, the KNN method will perform poorly because a point may not be able to find enough neighbors in range for proper estimation.

We find that a compressive sensing based algorithm, namely *Sparsity Rank-Singular Value Decomposition* (SRSVD) [10] can solve the problem of recovering absent data. The challenge is how to model our problem and leverage the hidden structure and redundancy of the collected data. We use the merging matrix [10] to merge and arrange all the RSSI values collected at different locations and at different times. In order to simplify the analysis, we use the rank of a matrix to judge the sparsity in compressive sensing. We formulate the problem as an optimization problem and try to find solutions.

SRSVD is a mathematical method to sparsity matrices. However, the characteristic structure of a merging matrix tends to be uncertain, due to the complex indoor environments. In reality, it is possible that several columns or rows are absent in the matrix at the same time, in which case SRSVD delivers quite low performance. We combine the SRSVD algorithm with the *K*-Nearest Neighbor (KNN) algorithm to address the problem. The approach interpolates only one element in an absent column or row and recovers the rest of absent data in the sparse matrix.

The major contributions of this paper are as follows. First, we propose to use the merging matrix to find the hidden structure and redundancy characteristics of the fingerprints. Second, we propose a

☆ A preliminary version of this article appeared in *Proceedings of the Eleventh IEEE Wireless Communications and Networking Conference (IEEE WCNC 2013)* [1].

* Corresponding author. Tel.: +86 755 8639 2325.
E-mail address: ai.chen@siat.ac.cn (A. Chen).

http://dx.doi.org/10.1016/j.comcom.2015.09.022
0140-3664/

novel approach based on compressive sensing to recover the absent data in the merging matrix. Third, we combine the SRSVD algorithm with the *K*-Nearest Neighbor (KNN) algorithm to deal with missing columns or rows in the matrix. Finally, we conduct experiments to evaluate the performance of the methods. The results show that given only half of the data, our approach can fully recover the fingerprint information with error rate below 6.6%. With merely 5% of the data, the approach can still recover the information with error rate below 14% and without loss of localization accuracy.

Organization: Section 2 introduces prior work on reducing fingerprint collection; Section 3 formulates the problem; Section 4 analyzes the problem and describes the solution; experiments are conducted in Section 5; and Section 6 concludes the paper.

2. Related work

One of the main methods of WiFi localization is based on RSSI. The related techniques can be classified into two categories: *propagation model based* and *fingerprint based*.

The propagation model based approach does not need signal fingerprints. Ubicarse [11] leverages Synthetic Aperture Radar (SAR) on hand-held devices that are twisted by their users along unknown paths. Ubicarse combines RF localization with stereo-vision algorithms to localize common objects with no RF source attached to them. The method in [12] leverages such a model to estimate the distance between the user and APs. Then it uses the extended Kalman filter to transform the distance to the user's position. The technique in [13] allows an organic positioning system to maintain its accuracy over time, based on outlier detection through clustering. A novel technique is proposed in [14], which uses the Gaussian Process Latent Variable Model (GPLVM) to relate RSS fingerprints and models of human movements (displacement, direction, etc.) as hidden variables. Utilizing a probabilistic RSS model derived from indoor signal propagation models that explicitly consider the effect of intervening walls and the building plan, the scheme in [15] first estimates the distance of the client to each of the APs and then obtains a location estimate through trilateration.

Radio propagation models are not very accurate for distance and position estimation, due to the multi-path effect and environmental interference. In comparison, the fingerprint based algorithms (e.g., [3,16]) normally have higher localization accuracy. In this approach, fingerprints collected with coordinate information are called labeled data and those without coordinates unlabeled data. The Label Propagation algorithm (LP-algorithm), by using semi-supervised learning in [8], tries to reduce the effort of collecting labeled data. In summary, these algorithms can reduce the work of labeling data, but still require collecting a large amount of unlabeled data.

In [17], a technique called the Signal-Distance Map (SDM) is proposed. SDM uses a truncated singular value decomposition technique to relate RSSI with geographical distance to the APs. Zee [18] and UnLoc [19] utilize WiFi and inertial sensors readings crowdsourced from users to build the fingerprint training set. In [20], a method called Walkie-Markie generates indoor pathway map by leveraging the locations of users when they pass WiFi marks. Furthermore, it uses the direction and distance information retrieved from the user trajectories to place the WiFi marks at real locations. Phaser [21] makes phased array signal processing practical on many WiFi access points deployed in the real world. In contrast, SpotFi [22] deployed on commodity WiFi infrastructure is able to achieve accuracy of 40 cm by calculating AoA of multipath components.

Compressive sensing techniques have been considered for reducing fingerprinting effort in a number of previous researches. In [23,24], the authors use l_1-minimization to solve the sparse signal recovery problem and use the map-adaptive Kalman filter to improve accuracy. Bayesian Compressive Sensing (BCS) based compressive sensing is used in [25,26]. In these techniques, the systems make full use of the relationship between the collected signals in the space. In [27], a multivariate Gaussian model is used to average the measurements of RSSI in the first step, and then compressive sensing is used to reduce the amount of information transmitted from a device in the second step. The Matrix Completion (MC) framework in [28] minimizes the number of RSSI fingerprints by sensing a subset of the available channels in a WiFi network. It provides a paradigm for reconstructing low-rank data matrices from a small number of randomly sampled entries. In the project, Environmental Space Time Improved Compressive Sensing (ESTI-CS) [29], real sensory data from the Intel Indoor, GreenOrbs, and Ocean Sense data sets are analyzed using the Multi-Attribute Assistant (MAA) component for data reconstruction.

Apart from RSSI-based methods, there are some other techniques of localization that are often used in combination with WiFi. Proximity detection is perhaps the simplest localization method. In such a method, the device estimates its location by simply detecting nearby radio sources [30]. The triangulation method provides improved localization accuracy by measuring the device's distance to multiple reference points [31,32]. When radio signal is missed during the user's navigation, dead reckoning is often used to fill in the gaps. Dead reckoning is a process of estimating the current position based on last determined position and incrementing that position based on known or estimated speeds over elapsed time [33]. For improved accuracy on real maps, the map matching techniques can be used. They include topological analysis, pattern recognition, or advanced techniques such as hierarchical fuzzy inference algorithms [19,34,35].

3. Problem formulation

This section describes the localization model and formulates the problem of recovering absent fingerprint data.

3.1. The system model

Normally, a WiFi-based localization system works in two phases: offline phase and online phase.

Offline phase: The indoor region of interest is divided into small grids. At the center point of each grid (reference location), an RF receiver collects the RSSI of each pre-deployed AP. At each reference location, the ID, coordinates, as well as each AP's RSSI, are recorded. These three elements together are called a *fingerprint*).

Online phase: The localization system estimates a device's location by comparing the measured RSSI against the fingerprints, and finding the closest matches.

3.2. Problem statement

In the offline phase, the device collects W RSSI values at each of the N reference location. Every time it measures the RSSI of M APs. Let $\mathbf{X}$ be the merging matrix of fingerprints with dimensions $W \cdot M \times N$. Also $\mathbf{X} = [\mathbf{E}_1\ \mathbf{E}_2 \ldots \mathbf{E}_W]^{\mathrm{T}}$, where $(\bullet)^{\mathrm{T}}$ is the transpose of a matrix, $\mathbf{E}_w$ the wth $(w = 1, 2, \ldots, W)$ sub-matrix of $\mathbf{X}$, and $\mathbf{E}_w(m, n)$ the RSSI of the mth $(m = 1, 2, \ldots, M)$ AP at the nth $(n = 1, 2, \ldots, N)$ location. We use an indicate matrix, $\mathbf{A}$, and a measurement matrix, $\mathbf{B}$, to represent the problem.

$$\mathbf{B} = \mathbf{A}. * \mathbf{X}$$
$$\mathbf{A} = [\mathbf{A}(i, j)] = \begin{cases} 0, & \text{if } \mathbf{B}(i, j) \text{ is absent} \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

where the symbol '.∗' denotes the dot multiplication, which means the multiplication of the corresponding elements in the matrix. The zero elements in matrix $\mathbf{A}$ mean that the corresponding data elements in matrix $\mathbf{B}$ are absent.

Designing an algorithm to recover the fingerprint matrix $\mathbf{X}$ based on the measurement matrix $\mathbf{B}$ with the absent elements can reduce the efforts of data collection.

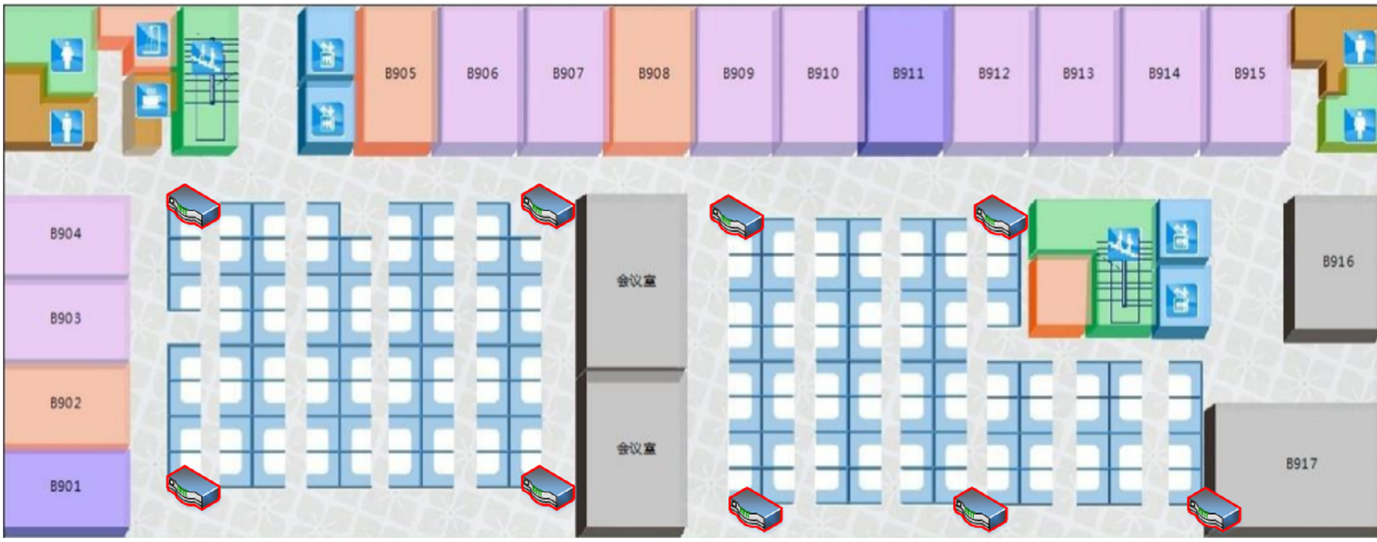

Fig. 1. The floor plan of experimental field.

4. Problem analysis and algorithm design

The theoretical foundation of our system is compressive sensing, which enables reduction of fingerprint collection if the fingerprint data is structured, i.e., can be sparsely represented. Existing works [36] have identified that the RSSI values from an AP are highly autocorrelated, which implies that the fingerprint data can be sparsely represented.

Since the fingerprint data are represented by a matrix, the *principal component analysis* (PCA) [37] provides a way of revealing the hidden structure in the matrix. As the singular value decomposition (SVD) is a common tool to implement the PCA, we utilize SVD to discover the hidden structure. Based on the theory of compressive sensing, the problem of recovering absent data in the matrix is converted into an optimization problem. In the following, the concept of SVD is first introduced, followed by a description of compressive sensing and its application to WiFi localization. Finally, a compressive sensing based solution is proposed.

4.1. Singular value decomposition

Singular value decomposition (SVD) [38] decomposes a given $a \times b$ (denoted as $W \cdot M \times N$) real matrix $\mathbf{X}$ into three matrices as follows.

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}} \tag{2}$$

where $\mathbf{U}$ is an $a \times a$ unitary matrix (i.e., $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{U}\mathbf{U}^{\mathrm{T}} = \mathbf{I}$, $\mathbf{I}$ is the unit matrix), and $\mathbf{V}$ is a $b \times b$ unitary matrix, $\mathbf{D}$ is an $a \times b$ diagonal matrix containing the non-zero singular values $\{\sigma_i | i = 1, 2, \ldots, r\}$ of $\mathbf{X}$, and r is the rank of $\mathbf{X}$. The singular values are sorted in descending order $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$. Since $\mathbf{D}$ is a diagonal matrix, Eq. (2) is rewritten as,

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}} = \sum_{i=1}^{\min(n,m)} \sigma_i u_i v_i^{\mathrm{T}} \tag{3}$$

where u_i and v_i are the columns of $\mathbf{U}$ and $\mathbf{V}^{\mathrm{T}}$, respectively. Let $\hat{\mathbf{X}}$ denote an approximation of $\mathbf{X}$ keeping only the r^* largest singular values in $\mathbf{X}$, as follows.

$$\hat{\mathbf{X}} = \sum_{i=1}^{r^*} \sigma_i \mathbf{Y}_i^{\mu}. \tag{4}$$

The above $\hat{\mathbf{X}}$ is the best rank-r^* approximation, where $r^* \leq \min(n, m)$, $\mathbf{Y}_i^{\mu} = u_i \times v_i^{\mathrm{T}}$, and the rank of $\mathbf{Y}_i^{\mu}$ is 1. Seeking solutions of $\hat{\mathbf{X}}$ is an optimization problem formulated as,

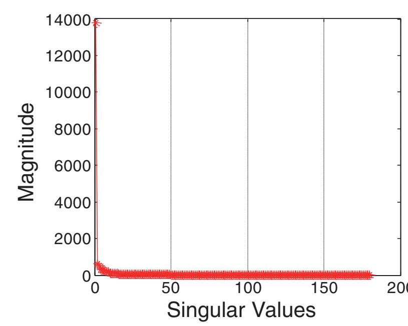

Fig. 2. The magnitude of the singular values.

$$\begin{aligned} &\text{minimize} && \|\mathbf{X} - \hat{\mathbf{X}}\|_F \\ &\text{subject to} && \operatorname{rank}(\hat{\mathbf{X}}) \leq r \end{aligned} \tag{5}$$

where $\|\cdot\|_F$ is the Frobenius norm, i.e., ($\|\mathbf{X}\|_F \triangleq \sqrt{\sum_{i,j} \mathbf{X}(i,j)^2}$)

Our experiment shows that SVD can be used to analyze the hidden structure of the merging matrix of fingerprints. We conduct our experiment in our office building, as depicted in Fig. 1, where the whole floor is divided into 52 reference locations (grids), each associated with four orientations. In each location, 20 RSSI measurements of 9 WiFi APs are recorded by a Motorola ME722 smartphone. So, the dimension of the merging matrix is 180×208. The singular value of each singular vector (σ_i, $i = 1, 2, \ldots, 180$) associated with the merging matrix is depicted in Fig. 2, where the magnitude of each singular value is proportional to the information entropy of the corresponding singular vector.

Fig. 2 shows that the matrix's information is primarily contained in a few principal components, since the largest singular value is much larger than the others. It suggests that the matrix will have a good approximation when r is small according to Eq. (5). Thus, the merging matrix can be sparse or compressible, which means collecting a few fingerprints can safely recover most information of the merging matrix.

4.2. Compressive sensing

The compressive sensing theory provides a mathematical foundation based on which a sparse collection of structured or redundant

Please cite this article as: Z. Gu et al., Reducing fingerprint collection for indoor localization, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.022

data can be used to recover the missing data. More specifically, compressive sensing adopts a special non-adaptive linear matrix to transform the original signal to a sparse one so that the hidden structure of the original signal can be kept. Based on this transform, the original signal can be sparsely sensed with a frequency much smaller than the Nyquist frequency, and the sensed signal can be well restored through certain optimization methods.

A vector is sparse if most of its elements are zero or nearly zero. However, a matrix with a large number of zeros is not necessarily a sparse matrix; instead, a matrix is sparse if it has a small rank.

Intuitively, our problem can be regarded as a matrix-completion problem, where sparsely sampled matrix entries are utilized to complete the rest of the matrix entries through compressive sensing. Though the merging matrix loses its low-rank nature because of the multipath effect, it is a sparse matrix in essence. Thus, the rank is an adaptive condition to construct the matrix.

4.3. Algorithm design

The problem of recovering missing fingerprints is expressed in Eq. (1). The measurement matrix **B** and the indicate matrix **A** are known a prior in the process. The merging matrix **X** is a constraint in the recovery process. The adaptive rank *r* represents the sparsity of the merging matrix. Solving the problem in Eq. (5) is equal to solving the following optimization problem:

$$\begin{array}{ll}\text{minimize} & \text{rank}(\hat{\mathbf{X}}) \\ \text{subject to} & \mathbf{A}.*\hat{\mathbf{X}} = \mathbf{B}\end{array} \tag{6}$$

where $\hat{\mathbf{X}}$ is the matrix to be recovered. Minimizing its rank is difficult since it is a non-convex problem. To simplify the problem, SVD can be utilized,

$$\hat{\mathbf{X}} = \mathbf{UDV}^{\mathrm{T}} = \mathbf{LR}^{\mathrm{T}} \tag{7}$$

where $\mathbf{L} = \mathbf{UD}^{1/2}$ and $\mathbf{R} = \mathbf{VD}^{1/2}$, which enable **L** and **R** to share common features such as dimension and rank. Previous works [10,38,39] have shown that the optimization problem formulated in formula (6) can be simplified under certain conditions. Specifically, when the restricted isometry property holds [39], minimizing the nuclear norm can achieve a minimum rank for a low-rank matrix. Formally, the equivalent optimization problem associated with the formula (6) can be formalized with the optimization objective to find **L** and **R** with low ranks by minimizing the summation of their Frobenius norms, as follows:

$$\begin{array}{ll}\text{minimize} & \|\mathbf{L}\|_F^2 + \|\mathbf{R}\|_F^2 \\ \text{subject to} & \mathbf{A}.*(\mathbf{LR}^{\mathrm{T}}) = \mathbf{B}.\end{array} \tag{8}$$

In practice, it might not be possible to find accurate **L** and **R** under the constraint of formula (8) due to three reasons. First, multipath interferences exist in WiFi signals so that strict satisfaction may lead to overfit of the approximation. Second, the merging matrix itself is not strictly low rank. Third, the collected RSSI values are recorded as integer values, which are not accurate enough themselves. Therefore, an optimization method for formula (8) is proposed as follows:

$$\text{minimize}\|\mathbf{A}.*(\mathbf{LR}^{\mathrm{T}}) - \mathbf{B}\|_F^2 + \lambda(\|\mathbf{L}\|_F^2 + \|\mathbf{R}\|_F^2) \tag{9}$$

Let $x = \|\mathbf{A}.*(\mathbf{LR}^{\mathrm{T}}) - \mathbf{B}\|_F^2$, $y = \lambda(\|\mathbf{L}\|_F^2 + \|\mathbf{R}\|_F^2)$, and $c = x + y$. Formula (9) can be rewritten as

$$\min(c). \tag{10}$$

The above solution seeks a low-rank approximation, but it does not strictly enforce the constraint of Eq. (8). The parameter λ is a factor that balances between *x* and *y*. The above approach is referred to as Sparsity Rank Singular Value Decomposition (SRSVD), which exploits both the sparsity nature of the fingerprints and the low-rank property of the SVD method.

We adopt the *Lagrange multiplier method* [10] to solve the optimization problem in formula (9). Note that the matrices **L** and **R** are two independent unknown variables in Eq. (9). It is difficult to obtain two unknown variables through one equation. To address this issue, an iteration method is adopted as follows: first, **L** and **R** are initialized randomly, and one of them is fixed to obtain the optimal solution for the other; second, their roles are switched to optimize the other with the same procedure; the above iteration repeats until a minimum *c* is determined.

Due to the nature of Frobenius norms, variables *x* and *y* are greater than or equal to 0. Assume that the variables *x* and *y* are equal to 0. Then, the following expression is obtained,

$$\begin{pmatrix}\mathbf{A}.*(\mathbf{LR}^{\mathrm{T}}) \\ \mathbf{R}^{\mathrm{T}}\end{pmatrix} = \begin{pmatrix}\mathbf{B} \\ \mathbf{0}\end{pmatrix} \tag{11}$$

A contradiction exists in Eq. (11). According to the equation, $\mathbf{R}^{\mathrm{T}}$ is equal to 0. Thus, **B** is also 0. However, it can be used to obtain the solution of $\mathbf{R}^{\mathrm{T}}$. In order to get a minimal *c*, two parameters λ and *r* are introduced: λ is the tradeoff factor and *r* is the upper bound of $\hat{\mathbf{X}}$'s rank. From the analysis above, the matrix $\hat{\mathbf{X}}$ can be well approximated if *r* is close to the rank of **X**. At the same time, *r* is less or equal to the rank of **L** and **R**.

The estimation error is introduced to control the iteration loop of the algorithm. The values of the parameters λ and *r* have a direct impact on the algorithm's performance. Thus, it is necessary to bound them. Since the merging matrix is structured and redundant, the rank *r* should not be large even though the matrix has a large dimension. The lower bound of *r* obviously should be larger than 1, while its upper bound should be a small integer. The tradeoff factor λ is not easy to determine. However, the order of magnitude can be first estimated as the bound, which can be repeatedly narrowed down through picking the median of the bound until a proper λ is identified that can minimize Ω (defined in Section 5).

The pseudocode of the solution discussed above is presented in Algorithm 1. Since the exact values of λ and *r* in Algorithm 1 are unknown, the algorithm searches for suitable values. In order to reduce the search time of λ, we select the median value of the bound and refresh it every time. Following the singular value decomposition, we use the minimum least square method to solve Eq. (11) in SVD.

4.3.1. Combining the K-Nearest Neighbor algorithm

Due to environmental complexity, there might exist several empty columns or rows in the merging matrix. Algorithm 1 recovers these columns or rows by filling them with zeros to keep the rank of the matrix low. Therefore, we introduce the *K*-Nearest Neighbor (KNN) algorithm to recover at least one element for each of empty columns or rows.

The main idea of the KNN algorithm is to find *K* nearest neighbors to recover the missing data, and the nearest neighbors are restricted in a circle with radius *d* (set to 10 m in this article). To recover an absent data element $\mathbf{X}(i, j)$ in the merging matrix, the algorithm first finds *K* nearest neighbors in the measurement matrix **B** by calculating geographical distance. Then, the average value of these neighbors is assigned as the estimate of $\mathbf{X}(i, j)$.

For an empty column or row, we first use the KNN algorithm to recover one element. When there is no empty column or row left, we then execute Algorithm 1 to recover the remaining data elements in the matrix. The details are described in Algorithm 2.

5. Experiments and results

In this section, we first provide a metric to estimate the error rate of the recovered data. To further evaluate the performance of the

Algorithm 1 SRSVD.

Input: r_l , r_u : lower and upper bounds of the rank.
λ_l , λ_u : lower and the upper bounds of the tradeoff factor.
τ , ω : number of iterations and the termination condition for λ.
$X_{m\times n}$, B : measurement matrix and indication matrix.
c , Ω : variable define in Eq. (9) and Section 5, respectively
Output: $\hat{X}$: recovered matrix.

for $r \leftarrow r_i$ to r_j do **do**
  X_l ,$\Omega_l \leftarrow$ SVD$(r, \lambda_l, \tau, X, B)$,$\Omega_{min} \leftarrow \Omega_l$;
  X_u ,$\Omega_u \leftarrow$ SVD$(r, \lambda_u, \tau, X, B)$;
  $\lambda_{next} \leftarrow$ median(λ_l, λ_u);
  while $(|\lambda_{next} - \lambda_{prev}| < \omega)$ **do**
    $X_{\lambda_{next}}, \Omega_{\lambda_{next}} \leftarrow$ SVD$(r, \lambda_{next}, \tau, X, B)$;
    $\hat{X}_{min}, \Omega_{min} \leftarrow \min(X_l, X_u, X_{\lambda_{next}}),\ \min(\Omega_l, \Omega_u, \Omega_{r_{next}})$;
    $\lambda_{prev} \leftarrow \lambda_{next}$;
    $\lambda_{next} \leftarrow$ select two min of Ω_l, Ω_u, Ω_{min}, confirm new $\lambda_{l_{new}}, \lambda_{u_{new}}$,
      median$(\lambda_{l_{new}}, \lambda_{u_{new}})$;
return $\hat{X}$;

Function SVD(r,λ,τ,**X**,**B**)
  $L \leftarrow$ random(L);
  for $i \leftarrow 1$ to τ do **do**
    $R \leftarrow$ MINLEASTSQUARE$([L; \sqrt{\lambda}I], [X; 0])^T$;
    $L \leftarrow$ MINLEASTSQUARE$([R^T; \sqrt{\lambda}I], [X; 0])^T$;
    if $c < c_{next}$ **then**
      $L_{next} \leftarrow L$; $R_{next} \leftarrow R$; $c_{next} \leftarrow c$;
  return $\hat{X} \leftarrow L * R^T$ and Ω (compute based on the definition);
Function MINLEASTSQUARE(A,B)
  $Y \leftarrow A^T A \backslash AB$;
  return Y;

Algorithm 2 SRSVD+KNN.

Input: r_l , r_u : lower and upper bounds of the rank.
λ_l , λ_u : lower and the upper bounds of tradeoff factor.
τ , ω : number of iterations and termination condition for λ.
$X_{m\times n}$, B : measurement matrix and indication matrix.
c , Ω : variables defined in Eq. (9) and Section 5
Output: $\hat{X}_{m\times n}$:Recovered matrix.

for $i \leftarrow 1$ to m **do**
  if $row[i]$ is absent **then**
    $j \leftarrow$ random$(1:n)$;
    $X(i, j) \leftarrow$ average {neighbors of $X(i, j)$};
for $j \leftarrow 1$ to n **do**
  if $column[j]$ is absent **then**
    $i \leftarrow$ random$(1:m)$;
    $X(i, j) \leftarrow$ average {neighbors of $X(i, j)$};
$\hat{X}_{m\times n} \leftarrow$SRSVD$(r_l, r_u, \lambda_l, \lambda_u, \tau, \omega, \mathbf{X}, \mathbf{B}, c, \Omega)$;
return $\hat{X}_{m\times n}$;

recovering algorithm, we compare two algorithms: KNN, which uses the mean of K nearest neighbors to interpolate, and Sparse Linear Algebra (SLA) [40], which interpolates NaN elements in a 2-d array using non-NaN elements. SLA is investigated for image recovery in [41], where the problem shares some common features with WiFi fingerprint recovery. The performance of the SLA recovery depends on how columns are related to each other. Finally, the *WKNN localization algorithm* [4] is employed to evaluate the localization accuracy based on the recovered fingerprints.

Table 1
Merging matrices of various sizes.

Index	W	M	N	Final size
1	−20	9	208	180 × 208
2	+20	9	208	9 × 4160
3	+208	9	20	9 × 4160
4	−208	9	20	1872 × 20
5	+9	20	208	20 × 1872
6	−9	20	208	180 × 208
7	+208	20	9	20 × 1872
8	−208	20	9	4160 × 9
9	+20	208	9	208 × 180
10	−20	208	9	4160 × 9
11	+9	208	20	208 × 180
12	−9	208	20	1872 × 20

5.1. Error analysis methodology

We propose to use the normalized difference between the recovered matrix $\hat{\mathbf{X}}$ and the original matrix $\mathbf{X}$ as a metric to measure the error rate. The difference between the two matrices is defined as the sum of the absolute values of the difference of the corresponding elements in these two matrices, i.e.

$$\sum_{i,j:\mathbf{A}(i,j)==0} |\mathbf{X}(i,j) - \hat{\mathbf{X}}(i,j)| \tag{12}$$

Thus, the normalized difference (error rate) is

$$\Omega = \frac{\sum_{i,j:\mathbf{A}(i,j)==0} |\mathbf{X}(i,j) - \hat{\mathbf{X}}(i,j)|}{\sum_{i,j:\mathbf{A}(i,j)==0} |\mathbf{X}(i,j)|} \tag{13}$$

5.2. Performance of merging matrix

In Section 3, the size of the merging matrix of the fingerprints $\mathbf{X}$ is $W \cdot M \times N$, where W, M, N refers to the collecting times at each location, the amounts of APs and the number of locations, respectively. However this is not the only way to construct the merging matrix. Since SVD can be used to analyze the a matrix's hidden structure, different structures of a matrix make different contributions to its inner correlations, which will further influence the effectiveness of the merging matrix. Hence, it is desirable to find a suitable matrix that can be recovered most effectively.

To make it easy to understand, we retain the major structure of $W \cdot M \times N$, but change its meaning. The fingerprint data contains the amounts of APs, geographic reference locations, and the times of recordings at each location, which are set as 9, 208, and 20, respectively. The sign of W indicates the direction ("+" for horizontal and "−" for vertical) of generating the merging matrix. For example, if $W \cdot M \times N$ is $+20 \cdot 9 \times 208$, then it means $\mathbf{X} = [\mathbf{E}_1\ \mathbf{E}_2 \ldots \mathbf{E}_{20}]$, where $\mathbf{E}_w$ is the wth ($w = 1, 2, \ldots, 20$) submatrix of $\mathbf{X}$ with size 9×208. We list all possible dimensions of the merging matrix in Table 1. We will evaluate their performance with different sizes in Section 5.2.

Out of the 12 ways to compose the merging matrix, we need to find the one with the best performance. In Fig. 3, the x-axis represents sample rate (i.e., $1 - \eta$), and the y-axis is the magnitude of Ω. The error rates of these 12 matrices with different structures are mostly influenced by the matrices' final sizes, despite the transpose relations between them. Matrices with final size 9×4160 (or 4160×9) perform worst. When the sample rate is more than 18%, matrices of size 180×208 or 208×180 perform a little worse than those with size 20×1872 or 1872×20, where the difference is negligible. At the same time it performs much better than the latter at a sample rate below 18%. For the sake of a lower sample rate, which means less effort of collection, to generate the merging matrix with size 180×208 or 208×180 is better than others, so that the structure of $-20{\cdot}9 \times 208$ is desirable.

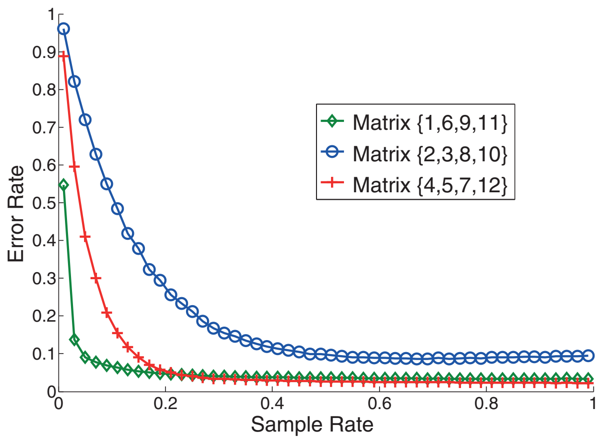

Fig. 3. Error rates of matrices with different structures.

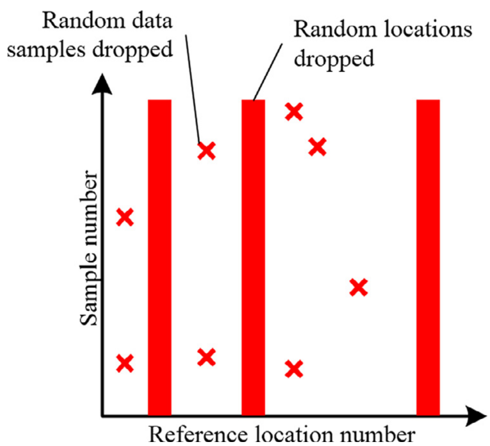

Fig. 4. HybridRandLoss sampling model.

5.3. Data preparing

We collect a complete set of fingerprints to compose the merging matrix using a Motorola smartphone. The detail has been described in Section 3.1. In order to evaluate the performance of different recovering algorithms, we should sample the merging matrix at a certain rate to obtain the relevant matrices for experiments. The process of fingerprint collection is as follows: one stands at a reference point with a mobile phone in hand. The device records the RSSI of all available APs at once. We compare the following models to simulate fingerprint reduction.

1. *SimpleRandLoss*: This is a simple random loss model. Due to the complexity of indoor environments, data points are dropped independently at random with probability η (loss rate).
2. *BlockRandLoss:* In this model, data are lost at random times (e.g., during sampling or transmission). In these cases, we may lose some random parts of the merging matrix X. We simulate the losses by discarding some of the blocks in X randomly, and dropping independent data points randomly with probability η.
3. *HybridRandLoss:* This model simulates a set of loss events, in which some (less than half) random reference locations are omitted when collecting fingerprints. At the same time some other data samples are missed due to sampling or transmission errors. We simulate the losses by dropping random columns and some random data points with probability η. Our algorithm primarily considers this model. The model is depicted in Fig. 4.

We make a comparison between algorithms for different loss models. Fig. 5 shows bar charts of the performance of the key algorithms for three models. It is obvious that SRSVD+KNN performs significantly better across all loss models. In reality, we reduce collecting fingerprint in the HybridRandLoss model. Specifically, the proposed algorithm will perform poorly when more than half of reference locations are omitted in the HybridRandLoss model.

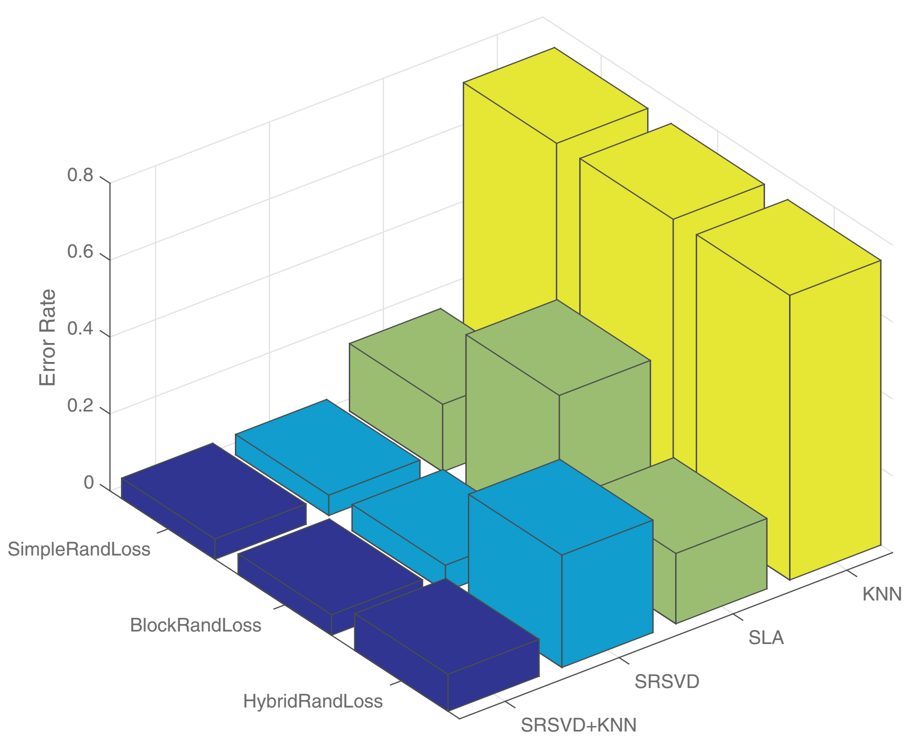

Fig. 5. Performance of three models with sample rate 0.2.

Please cite this article as: Z. Gu et al., Reducing fingerprint collection for indoor localization, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.022

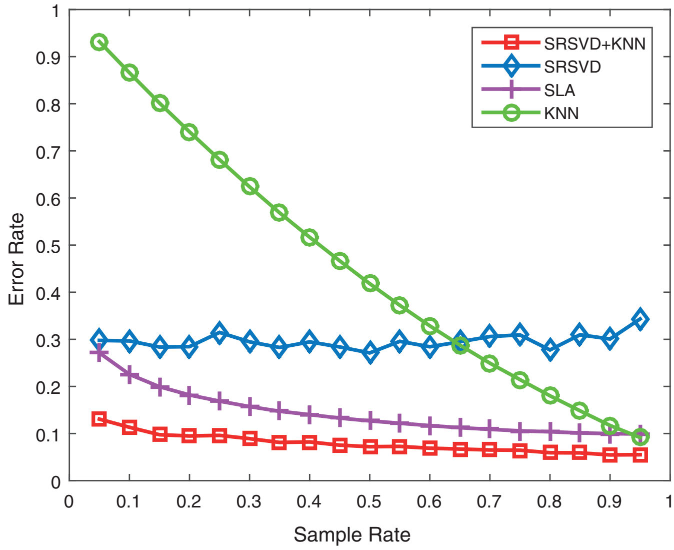

Fig. 6. Performance of different algorithms for the HybridRandLoss model.

5.4. Data recovering performance

We compare SRSVD+KNN with other recovery methods in the HybridRandLoss model as we show in Section 5.3. In the experiments, the sample rate increases from 5% to 95% with step size 5%. Fig. 6 shows the performance of each method. The higher the sample rate, the lower the error rate. When the sample rate is high, all the methods have similar performances. However, SRSVD+KNN performs significantly better than others at a low sample rate.

SLA works better than SRSVD or KNN. However, the combination of SRSVD and KNN turns out to perform very well. The results show that SRSVD+KNN uses only 5% of the original data to recover all the fingerprints with error rate less than 14%. When half of the original data are used, the error rate is within 6.6%.

5.5. Localization using recovered fingerprints

A set of experiments are conducted to evaluate the accuracy of the WKNN localization algorithm with recovered fingerprints, in comparison with the effect of the original complete fingerprints.

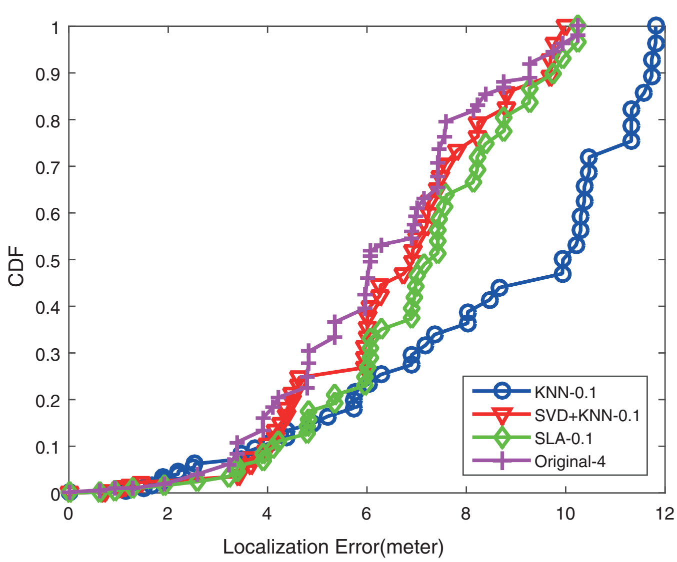

Fig. 7. CDF of different algorithms with sample rate 0.1 for the HybridRandLoss model.

The WKNN algorithm is a weighted K-Nearest Neighbor method. In the online phase, a user at an unknown location measures the RSSI values. Then the WKNN algorithm finds the best K matched locations in the fingerprints. Finally, the user is localized by calculating the weighted average of the K neighbors' coordinates.

In the experiment, RSSIs are measured at 10 different locations with known coordinates, and the measurement at each location is repeated five times. The corresponding localization results and the associated cumulative distribution function (CDF) for WKNN ($K = 4$ proves to be a good choice) based on the complete fingerprints, the recovered fingerprints through KNN, SLA, and SRSVD+KNN are calculated.

Fig. 7 shows the performance of localization. SRSVD+KNN-0.1, SLA-0.1 and KNN-0.1 represent the results of localization based on recovered fingerprints through SRSVD+KNN, SLA and KNN with $\eta = 0.9$, respectively. Fig. 7 indicates that the curve of SRSVD+KNN-0.1 is close to the one that uses the original complete fingerprints.

6. Conclusion

In this paper, a novel approach has been proposed to reduce the measurement effort required for collecting WiFi fingerprints. All the collected data are merged into the merging matrix. The SVD method reveals the hidden structure and redundancy characteristics via the merging matrix, which makes it possible to apply the compressive sensing technique for data reduction. The challenge is how to recover the absent data in the merging matrix faithfully, while minimizing the effort of data collection. Experimental results show that using 5% of the original data, the proposed approach SRSVD+KNN can recover all the fingerprints with error rate less than 14%. The localization accuracy with the recovered fingerprints is similar to the one with the original complete fingerprints.

Acknowledgments

This work is supported in part by the Shenzhen Overseas High-level Talents Innovation and Entrepreneurship Funds under Grants KQC201109050097A and KQCX20140520154115026, the Shenzhen Basic Research Funds under Grant JCYJ20140610151856733, and the National Natural Science Foundation of China under Grant no. 60903222.

References

[1] Y. Zhang, Y. Zhu, M. Lu, A. Chen, Using compressive sensing to reduce fingerprint collection for indoor localization, in: Proceedings of the IEEE Wireless Communications and Networking Conference Workshops, IEEE WCNC, 2013, pp. 4540–4545.

[2] J.K.-Y. Ng, K.-Y. Lam, Q.J. Cheng, K.C.Y. Shum, An effective signal strength-based wireless location estimation system for tracking indoor mobile users, J. Comput. Syst. Sci. 79 (7) (2013) 1005–1016.

[3] P. Bahl, V.N. Padmanabhan, Radar: an in-building RF-based user location and tracking system, in: Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2000, 2, IEEE, 2000, pp. 775–784.

[4] P. Bahl, V.N. Padmanabhan, A. Balachandran, Enhancements to the RADAR User Location and Tracking System, Technical Report, Microsoft Research, 2000.

[5] A. Smailagic, D. Kogan, Location sensing and privacy in a context-aware computing environment, IEEE Wirel. Commun. 9 (5) (2002) 10–17.

[6] A. Haeberlen, E. Flannery, A.M. Ladd, A. Rudys, D.S. Wallach, L.E. Kavraki, Practical robust localization over large-scale 802.11 wireless networks, in: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, ACM, 2004, pp. 70–84.

[7] A.M. Ladd, K.E. Bekris, A. Rudys, L.E. Kavraki, D.S. Wallach, Robotics-based location sensing using wireless ethernet, Proceedings of the International Conference on Mobile Computing and Networking, ACM MobiCom, 2004.

[8] T. Pulkkinen, T. Roos, P. Myllymäki, Semi-supervised learning for wlan positioning, in: Proceedings of International Conference on Artificial Neural Networks and Machine Learning – ICANN 2011, Springer, 2011, pp. 355–362.

[9] S. Liu, H. Luo, S. Zou, A low-cost and accurate indoor localization algorithm using label propagation based semi-supervised learning, in: Proceedings of the Mobile Ad-hoc and Sensor Networks, MSN, 2009, pp. 108–111.

[10] Y. Zhang, M. Roughan, W. Willinger, L. Qiu, Spatio-temporal compressive sensing and internet traffic matrices, in: Proceedings of the ACM SIGCOMM Computer Communication Review, 39, ACM, 2009, pp. 267–278.
[11] S. Kumar, S. Gil, D. Katabi, D. Rus, Accurate indoor localization with zero start-up cost, in: Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, MobiCom, ACM, 2014, pp. 483–494.
[12] A. Kotanen, M. Hannikainen, H. Leppakoski, T. Hamalainen, Positioning with IEEE 802.11 b wireless LAN, in: Proceedings of the 14th IEEE Personal, Indoor and Mobile Radio Communications, PIMRC, 3, IEEE, 2003, pp. 2218–2222.
[13] J.-g. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, J. Ledlie, Growing an organic indoor location system, in: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, ACM, 2010, pp. 271–284.
[14] B. Ferris, D. Fox, N.D. Lawrence, WiFi-SLAM using Gaussian process latent variable models, in: Proceedings of International Joint Conference on Artificial Intelligence, IJCAI, 7, 2007, pp. 2480–2485.
[15] M. Robinson, I. Psaromiligkos, Received signal strength based location estimation of a wireless LAN client, in: Proceedings of 2005 IEEE Wireless Communications and Networking Conference, 4, IEEE, 2005, pp. 2350–2354.
[16] S. He, S.-H. G. Chan, L. Yu, N. Liu, Fusing noisy fingerprints with distance bounds for indoor localization, in: Proceedings of the 34th Annual IEEE International Conference on Computer Communications, INFOCOM, 2015, pp. 2506–2514.
[17] H. Lim, L.-C. Kung, J.C. Hou, H. Luo, Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure, Wirel. Netw. 16 (2) (2010) 405–420.
[18] A. Rai, K.K. Chintalapudi, V.N. Padmanabhan, R. Sen, Zee: zero-effort crowdsourcing for indoor localization, in: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, ACM, 2012, pp. 293–304.
[19] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, R.R. Choudhury, No need to war-drive: unsupervised indoor localization, in: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, ACM, 2012, pp. 197–210.
[20] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, Y. Zhang, Walkie-Markie: indoor pathway mapping made easy, in: Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation, USENIX Association, 2013, pp. 85–98.
[21] J. Gjengset, J. Xiong, G. McPhillips, K. Jamieson, Phaser: enabling phased array signal processing on commodity WiFi access points, in: Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, MobiCom, ACM, 2014, pp. 153–164.
[22] M. Kotaru, K. Joshi, D. Bharadia, S. Katti, SpotFi: decimeter level localization using WiFi, in: Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, SIGCOMM, ACM, 2015, pp. 269–282.
[23] C. Feng, W.S.A. Au, S. Valaee, Z. Tan, Received-signal-strength-based indoor positioning using compressive sensing, IEEE Trans. Mobile Comput. 11 (12) (2012) 1983–1993.
[24] A.W.S. Au, C. Feng, S. Valaee, S. Reyes, S. Sorour, S.N. Markowitz, D. Gold, K. Gordon, M. Eizenman, Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device, IEEE Trans. Mobile Comput. 12 (10) (2013) 2050–2062.
[25] D. Milioris, G. Tzagkarakis, P. Jacquet, P. Tsakalides, Indoor positioning in wireless LANs using compressive sensing signal-strength fingerprints, in: Proceedings of European Signal Processing Conference, EUSIPCO11, Barcelona, Spain, 2011, pp. 1776–1780.
[26] R. Nandakumar, K.K. Chintalapudi, V.N. Padmanabhan, Centaur: locating devices in an office environment, in: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, MobiCom, ACM, 2012, pp. 281–292.
[27] D. Milioris, G. Tzagkarakis, A. Papakonstantinou, M. Papadopouli, P. Tsakalides, Low-dimensional signal-strength fingerprint-based positioning in wireless LANs, Ad Hoc Netw. 12 (2014) 100–114.
[28] S. Nikitaki, G. Tsagkatakis, P. Tsakalides, Efficient training for fingerprint based positioning using matrix completion, in: Proceedings of the 20th European Signal Processing Conference, EUSIPCO, IEEE, 2012, pp. 195–199.
[29] L. Kong, M. Xia, X.-Y. Liu, G. Chen, Y. Gu, M.-Y. Wu, X. Liu, Data loss and reconstruction in wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 25 (11) (2014) 2818–2828, doi:10.1109/TPDS.2013.269.
[30] Y. Tian, R. Gao, K. Bian, F. Ye, T. Wang, Y. Wang, X. Li, Towards ubiquitous indoor localization service leveraging environmental physical features, in: Proceedings of IEEE INFOCOM, IEEE, 2014, pp. 55–63.
[31] S. Sen, J. Lee, K.-H. Kim, P. Congdon, Avoiding multipath to revive inbuilding WiFi localization, in: Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys, ACM, 2013, pp. 249–262.
[32] F. Wen, C. Liang, Fine-grained indoor localization using single access point with multiple antennas, IEEE Sens. J. 15 (3) (2015) 1538–1544.
[33] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, F. Zhao, Travi-navi: Self-deployable indoor navigation system, in: Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, MobiCom, ACM, 2014, pp. 471–482.
[34] K. Chintalapudi, A. Padmanabha Iyer, V.N. Padmanabhan, Indoor localization without the pain, in: Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking, ACM, 2010, pp. 173–184.
[35] Z. Xiao, H. Wen, A. Markham, N. Trigoni, Lightweight map matching for indoor localisation using conditional random fields, in: Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, IPSN-14, IEEE, 2014, pp. 131–142.
[36] M. Youssef, A. Agrawala, The Horus location determination system, Wirel. Netw. 14 (3) (2008) 357–374.
[37] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemom. Intell. Lab. Syst. 2 (1) (1987) 37–52.
[38] E.J. Candes, B. Recht, Exact matrix competition via convex optimization, Found. Comput. Math. 9 (6) (2009) 717–772.
[39] B. Recht, M. Fazel, P.A. Parrilo, Guaranteed minimum rank solution to linear matrix equations via nuclear norm minimization, SIAM Rev. 52 (3) (2010) 471–501.
[40] J. DErrico, inpaint-nans. URL http://www.mathworks.com/matlabcentral/fileexchange/4551-inpaint-nans.
[41] A. Hughes, J. Hopgood, N. Robertson, Height approximation for audio source localisation and tracking, in: Proceedings of the 21st European Signal Processing Conference, EUSIPCO, 2013, pp. 1–5.