Research Article

International Journal of
Distributed
Sensor Networks

International Journal of Distributed Sensor Networks
2017, Vol. 13(8)

DOI: 10.1177/1550147717723805
journals.sagepub.com/home/ijdsn
SAGE

An efficient dictionary refinement algorithm for multiple target counting and localization in wireless sensor networks

Baoming Sun[1], Yan Guo[1], Gengfa Fang[2] and Eryk Dutkiewicz[2]

Abstract

Many applications provided by wireless sensor networks rely heavily on the location information of the monitored targets. Since the number of targets in the region of interest is limited, localization benefits from compressive sensing, sampling number can be greatly reduced. Despite many compressive sensing–based localization methods proposed, existing solutions are based on the assumption that all targets fall on a sampled and fixed grid, performing poorly when there are targets deviating from the grid. To address such a problem, in this article, we propose a dictionary refinement algorithm where the grid is iteratively adjusted to alleviate the deviation. In each iteration, the representation coefficient and the grid parameters are updated in turn. After several iterations, the measurements can be sparsely represented by the representation coefficient which indicates the number and locations of multiple targets. Extensive simulation results show that the proposed dictionary refinement algorithm achieves more accurate counting and localization compared to the state-of-the-art compressive sensing reconstruction algorithms.

Keywords

Wireless sensor networks, compressive sensing, dictionary refinement, counting, localization

Date received: 17 October 2016; accepted: 22 June 2017

Academic Editor: Antonio Lazaro

Introduction

Location is highly critical to many services provided by wireless sensor networks (WSNs),[1] such as geographic routing, wildlife monitoring, and health caring. Localization is a fundamental issue of WSNs that has been extensively studied in the literature. In daily life, the global positioning system (GPS)[2] is widely applied to achieve self-localization. However, there are still some situations where it does not work well (e.g. indoors or under the ground). Moreover, having each target GPS-equipped is extremely infeasible and expensive for WSNs.

The limitations of GPS have promoted a large body of localization approaches, which can be classified into two categories: range-based and range-free approaches. Range-based approaches are simple but susceptible to fading, noise, and non–line of sight. Range-free approaches perform localization by exploiting the connectivity between targets and sensors. They do not need extra hardware support but are usually imprecise and easily sensitive to the density of sensors. More

[1]College of Communications Engineering, PLA University of Science and Technology, Nanjing, China
[2]School of Computing and Communications, University of Technology Sydney (UTS), Ultimo, NSW, Australia

Corresponding author:
Yan Guo, College of Communications Engineering, PLA University of Science and Technology, No. 2 Biaoying, Yudao Road, Nanjing, 210007 Jiangsu Province, China.
Email: guoyan_1029@sina.com

seriously, both range-based and range-free approaches are computationally inefficient as they require the exchange of a large number of data to measure distances or determine connectivity. However, in most applications, sensors are resource-constrained (e.g. low power, low memory, and low operational ability), leading these solutions impractical. Therefore, it is quite necessary to develop a localization approach with less data collection and processing.

Compressive sensing (CS)[3,4] offers a novel solution to the localization problem in WSNs. As a novel signal processing technique, CS theory asserts that a small number of measurements will suffice for recovering the original sparse or compressible signals. Since the number of targets in the region of interest is limited, localization benefits from CS, sampling number can be greatly reduced. As a consequence, CS attracts considerable attention in the localization field, and a lot of localization approaches based on CS are proposed that will be discussed later. In these approaches, the continuous physical space is discretized into a fixed grid which corresponds to a dictionary. By assuming that all targets fall exactly on some grid points, the measurement vector can be sparsely represented as a sparse linear combination of the dictionary atoms. Then, localization is accomplished by sparse signal recovery followed by support detection.

Unfortunately, as a matter of fact, the assumption usually does not establish that all targets fall exactly on a fixed grid. It is noteworthy that there always exist targets that deviate from the fixed grid no matter how fine the space is sampled. In such a case, there exists mismatch between the assumed and actual sparsifying dictionaries. This is the so-called "dictionary mismatch" problem. Existing research indicates that the existence of dictionary mismatch will deteriorate the performance of CS dramatically.[5,6]

Since CS has been focused on the signals that can be sparsely or compressibly represented under a finite dictionary, discretization of the physical space is inevitable. It is intuitively reasonable that both dictionary mismatch and recovery error can be reduced with a dense grid. Therefore, one naturally wonders whether a denser grid leads to more accurate sparse signal recovery. In fact, according to CS theory, the sampled grid should not be too dense. This is because the dictionaries, corresponding to densely-sampled grids, have high inter-column correlation. The high correlation of dictionary atoms violates the restricted isometry property (RIP) condition of CS.[19] In fact, the performance of CS may degrade considerably in the presence of dictionary mismatch, even when the physical space is fine-grained discretized.

In this article, we study the dictionary mismatch problem and develop an efficient dictionary refinement algorithm. The key idea is to dynamically adjust the grid to alleviate or even eliminate the mismatch between the assumed and the actual sparsifying dictionaries. First of all, we regard the sparsifying dictionary as a parameterized dictionary, with the sampled grid as the underlying parameters. Consequently, the original counting and localization problem is formulated as a joint sparse signal recovery and grid parameter estimation problem. Then, an iterative two-step algorithm is developed that alternately optimizes over the sparse signal and grid parameters in a manner of optimizing over one while keep another fixed. Therefore, the original joint optimization problem is transformed into two sub-problems that can be effectively solved by existing optimization tools. Finally, we demonstrate via simulation that the proposed dictionary refinement algorithm performs significantly better than the state-of-the-art CS reconstruction algorithms.

The main contributions of this article can be summarized as follows:

- We study the dictionary mismatch problem in CS-based localization and develop an efficient dictionary refinement algorithm. Based on the algorithm, the grid can be dynamically adjusted to alleviate or even eliminate the mismatch between the assumed and actual sparsifying dictionaries.
- To achieve dictionary refinement, we view the sparsifying dictionary as a parameterized dictionary, with the sampled grid as the underlying parameters. Therefore, the original counting and localization problem is formulated as a joint sparse signal recovery and grid parameter estimation problem.
- To solve the joint optimization problem, we decompose it into two sub-problems and develop an iterative two-step algorithm to solve the sub-problems. In each iteration, the sparse signal and grid parameters are updated in turn.
- We conduct extensive simulations to evaluate the performance of the proposed dictionary refinement algorithm. The superiority of our algorithm compared with other sparse reconstruction algorithms is confirmed by the simulation results.

The rest of this article is organized as follows. Section "Related work" gives a review on related work. A brief background information about CS is presented in section "CS." Section "Network model and problem statement" provides the network model and problem statement. In section "Problem formulation," we mathematically formulate the counting and localization problem. Section "Localization via dictionary refinement" provides detailed descriptions on the proposed dictionary refinement algorithm. The performance of our algorithm is demonstrated in section "Numerical evaluation." Finally, conclusion is summarized in section "Conclusion."

Notations used in this article are as follows. $\mathbb{R}$ denotes the set of real numbers. Capital boldface letters and lowercase boldface letter are reserved for matrices and vectors, respectively. $|\cdot|$ denotes the amplitude of a scalar or cardinality of a set. $\times$ denotes Cartesian product. $\|\cdot\|_0$, $\|\cdot\|_1$, and $\|\cdot\|_2$ are the ℓ_0, ℓ_1, and ℓ_2 norms, respectively. For a given matrix $\mathbf{A}$, $\mathbf{A}^{-1}$ and $\mathbf{A}^{\mathrm{T}}$ denote the inverse and transpose, respectively.

Related work

In this section, we review the related work in the field of CS-based localization. Cevher et al.[7] propose to apply CS to target localization for the first time. The localization problem is transformed into a sparse recovery problem by sampling the physical space into a grid. In addition, a Bayesian framework[8] is proposed and the sparse approximation to its optimal solution is also provided. However, the drawback that a dictionary needs to be maintained at each sensor limits its applications. Meanwhile, the demand of communication among sensors also leads to poor performance in sparse WSNs. Feng et al.[9] model the locations of multiple targets as a sparse matrix and propose a CS-based indoor localization approach. Although it declares to be able to localize multiple targets, the approach can localize only one target once in that the data compression is not sufficient enough. In Feng et al.,[10,11] a clustering idea is further introduced to reduce system cost. The localization performance relies heavily on the clustered results and cluster matching algorithms. Nevertheless, in general, it is difficult to choose a proper clustering scheme and an effective cluster matching algorithm. Zhang et al.[12] consider multiple target localization without the prior knowledge of target number. Different from Feng et al.,[9] the authors view the locations of multiple targets as a sparse vector whose sparsity corresponds to the number of targets. Then, a greedy matching pursuit (GMP) algorithm is proposed to recover the sparse signal with a high probability. Au et al.[13] apply CS to develop a positioning system which consists of a coarse stage and a fine stage. The coarse stage is executed to obtain the approximate positions of a target using a proximity constraint and then more accurate position is obtained by CS in the fine stage. Guyen et al.[14] propose an indoor positioning system where the radio map is decomposed into a dictionary and a sparse representation matrix using the K-SVD dictionary learning algorithm. Then, the real-time received signal strength (RSS) vector is matched with the columns of the sparse representation matrix. The reference point with the least matching error is determined to be closest to the target. In fingerprint-based localization approaches, the construction of fingerprint database is usually time-costing. By exploiting the spatial and temporal relativity, Zhang et al.[15] develop a new fingerprint database construction method with only a few fingerprint collection. Different from the above-mentioned range-based methods, Liu et al.[16] develop a range-free CS-based localization method. Instead of collecting RSS measurements directly, the information whether targets are in the range of sensors are utilized to achieve localization at the cost of unreliable accuracy. Nguyen and Shin[17] develop a CS-based localization method which collects RSS measurements according to a deterministic sensing matrix rather than random sensing matrix. By exploiting the sparsity of spatial signals and the compressibility of temporal signals, Sun et al.[18] develop a two-dimensional (2D) localization framework using CS.

In the above-mentioned solutions, a common assumption is made that all targets fall exactly on a fixed grid. However, it should be noted that there always exist targets that deviate from the grid no matter how fine the space is sampled. In such a case, the traditional CS-based localization methods perform poor. Additionally, it should be noted that we assume each RSS measurement taken by a sensor is the sum of the strengths of the received signals that come from all targets while the sensor cannot distinguish the signals from different targets. Therefore, it is impossible to directly derive distance estimates or approximation information between sensors and targets. As a result, the traditional range-based or range-free localization methods cannot be applied in our context.

CS

Consider a discrete signal given by the vector $\mathbf{s} \in \mathbb{R}^N$. It is sparse if $\mathbf{s}$ has only a few non-zero coefficients, that is, $\|\mathbf{s}\|_0 \ll N$. More generally, $\mathbf{s}$ is compressible in the sense that it has many small coefficients, and only a few large coefficients. Results in CS have shown that if $\mathbf{s}$ is sparse or compressible, then it is possible to reconstruct it from $M (M \ll N)$ measurements yielded by a measurement matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$

$$\mathbf{z} = \mathbf{\Phi s} \tag{1}$$

However, it is worthy noting that few signals in practice are truly sparse or compressible themselves; rather they can be sparsely or compressibly represented as $\mathbf{s} = \mathbf{\Psi w}$ in some representation basis $\mathbf{\Psi} \in \mathbb{R}^{N \times N}$, where $\mathbf{w}$ is sparse or compressible. In such a case, the measurement vector can be reexpressed as

$$\mathbf{z} = \mathbf{\Phi \Psi w} = \mathbf{D w} \tag{2}$$

where $\mathbf{D}$ is named as a sparsifying dictionary, its column vector $\mathbf{d}_i$ is called as atom. Therefore, the sparse recovery problem can be transferred to a dictionary atom selection problem.

Given $\mathbf{z}$ and $\mathbf{D}$, the goal of CS is to recover $\mathbf{w}$ with high accuracy. Generally, CS achieves this by exploiting the sparsity (compressibility) property of $\mathbf{w}$. At present, there are various sparse recovery algorithms in the literature, for example, basis pursuit (BP),[20] orthogonal matching pursuit (OMP),[21] and sparse Bayesian learning (SBL).[22]

Network model and problem statement

Network model

Without loss of generality, we consider the problem of localization for K targets in a 2D area covered by a WSN, as can be seen in Figure 1(a). Each target is equipped with an emitter which broadcasts beacons periodically. The RSS measurements are collected by M sensors and then transferred to a distant fusion center (FC). At last, in the FC, a sparse reconstruction algorithm is applied to recover the locations of these targets. In this manner, the computing load is transferred from sensors to the FC, significantly reducing the energy consumption of sensors. Assume that the targets and sensors are located with positions $\{\boldsymbol{\tau}_k\}_{k=1}^K$ and $\{\mathbf{t}_m\}_{m=1}^M$, where $\boldsymbol{\tau}_k = (x_k, y_k)$ denotes the position of the kth target and $\mathbf{t}_m$ denotes the position of the mth sensor. Then, the measurement of the mth sensor can be given as

$$z_m = \sum_{k=1}^{K} \alpha_k f(\mathbf{t}_m, \boldsymbol{\tau}_k) + \varepsilon_m \tag{3}$$

where α_k, ε_m, and $f(\mathbf{t}, \boldsymbol{\tau})$ denote transmitted power, additive noise, and energy decay function, respectively. In this article, we use the path loss model presented in Clouqueur et al.[23] to define the energy decay function

$$f(\mathbf{t}, \boldsymbol{\tau}) = \begin{cases} 1, & \text{if} d < d_0 \\ \dfrac{1}{(d/d_0)^{\gamma}}, & \text{otherwise} \end{cases} \tag{4}$$

where $d = \|\mathbf{t} - \boldsymbol{\tau}\|_2$ denotes the distance between the transmitter and receiver; d_0 denotes the reference distance; and finally, γ denotes the path loss coefficient. For the sake of simplicity, we make an assumption that the transmitted powers of all targets are 1. Our goal is to estimate the number K and locations $\boldsymbol{\Gamma} = \{\boldsymbol{\tau}_k\}_{k=1}^K$ of these targets from the noisy measurement vector $\mathbf{z} = [z_1, z_2, \ldots, z_M]^{\mathrm{T}}$.

Problem statement

Since the number of targets is limited, localization benefits from CS, the measurement number can be greatly reduced. In order to implement CS, we divide the area of interest into a grid. The grid lines are denoted by vectors $\mathbf{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}}$ and $\mathbf{y} = [y_1, y_2, \ldots, y_n]^{\mathrm{T}}$, where x_i and y_i denote the ith grid lines in x-axis and y-axis, respectively. The intersections of grid lines are grid points. We number these grid points from 1 to N ($N = n^2$). The numbering rule is shown in Figure 1(b) where the blue numbers denote the number of different grid points. The positions of these grid points, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_i\}_{i=1}^N$, can be denoted by $\boldsymbol{\Theta} = \mathbf{x} \times \mathbf{y}$ and are known in advance. Define the M vector

$$\mathbf{d}(\boldsymbol{\theta}_i) = [f(\mathbf{t}_1, \boldsymbol{\theta}_i), f(\mathbf{t}_2, \boldsymbol{\theta}_i), \ldots, f(\mathbf{t}_M, \boldsymbol{\theta}_i)]^{\mathrm{T}} \tag{5}$$

as the atom formed by the ith grid point $\boldsymbol{\theta}_i$, where $f(\mathbf{t}_m, \boldsymbol{\theta}_i)$ denotes the RSS collected by the mth sensor

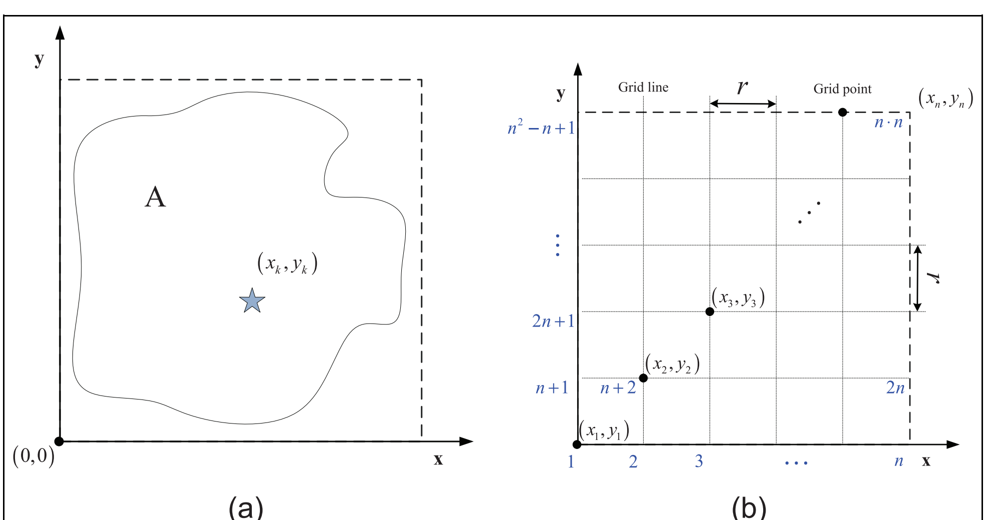

Figure 1. (a) The scenario of multiple target localization and (b) an example of grid numbering.

from the target at the ith grid point. Therefore, the sparsifying dictionary formed by all grid points $\boldsymbol{\Theta}$ is given as

$$\mathbf{D}(\boldsymbol{\Theta}) = [\mathbf{d}(\boldsymbol{\theta}_1), \mathbf{d}(\boldsymbol{\theta}_2), \ldots, \mathbf{d}(\boldsymbol{\theta}_N)] \tag{6}$$

Conventional localization approaches based on CS pre-suppose that all targets fall exactly on the discretized grid, namely, $\boldsymbol{\Gamma} \subset \boldsymbol{\Theta}$, then the measurement vector can be approximated as

$$\begin{aligned}\mathbf{z} &= \sum_{k=1}^{K} \alpha_k \mathbf{d}(\boldsymbol{\tau}_k) + \boldsymbol{\varepsilon} \\ &= \sum_{i=1}^{N} w_i \mathbf{d}(\boldsymbol{\theta}_i) + \boldsymbol{\varepsilon} \\ &= \mathbf{D}(\boldsymbol{\Theta})\mathbf{w} + \boldsymbol{\varepsilon}\end{aligned} \tag{7}$$

where $\mathbf{w} = [w_1, w_2, \ldots, w_N]^{\mathrm{T}}$ denotes the representation coefficient formed by approximately representing $\mathbf{z}$ in $\mathbf{D}(\boldsymbol{\Theta})$ and encodes the number and locations of considered targets. If the kth target falls on the ith grid point, then its element $w_i = \alpha_k$; otherwise, $w_i = 0$. Since the number of targets K is much smaller than the number of grid points N, $\mathbf{w}$ is a K-sparse signal. As a result, the localization issue becomes a sparse recovery issue and $M(M \ll N)$ measurements are sufficient to recover $\mathbf{w}$ by exploiting its sparsity with CS.

However, in general, the targets may not fall exactly on the discretized grid no matter how fine we discretize the physical space. The existence of off-grid targets leads to the mismatch between the assumed dictionary and the actual dictionary. As a matter of fact, in practice, a small dictionary mismatch may deteriorate the signal recovery performance dramatically.

Problem formulation

To address such a dictionary mismatch problem, an effective solution is to iteratively optimize over the grid parameter $\boldsymbol{\Theta}$ and the representation coefficient w, which can be mathematically formulated as

$$\left[\tilde{\mathbf{w}}, \tilde{\boldsymbol{\Theta}}\right] = \arg\min_{\mathbf{w}, \boldsymbol{\Theta}} \left\{ \|\mathbf{z} - \mathbf{D}(\boldsymbol{\Theta})\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_1 \right\} \tag{8}$$

where the regularization parameter λ controls the sparsity of the representation coefficient $\mathbf{w}$.

Initially, the continuous space is discretized as a uniform grid $\mathbf{x}^{(0)} \times \mathbf{y}^{(0)}$ with r denoting the distance between two adjacent grid lines. Then, the grid lines are dynamically adjusted to alleviate dictionary mismatch by optimizing grid parameters $\mathbf{x}$ and $\mathbf{y}$. We make an assumption that the distance between any two targets is no less than r. Based on the assumption, the off-grid

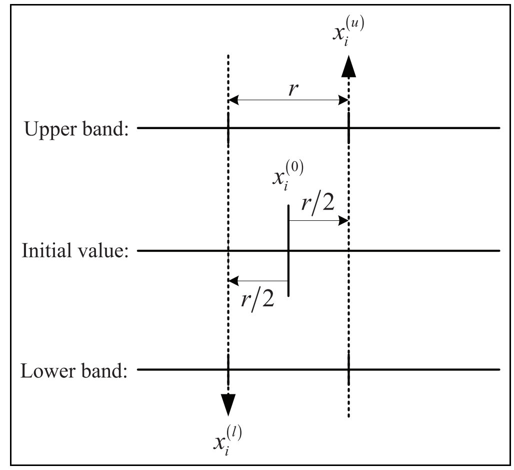

Figure 2. An illustration of the upper and lower bounds of the x-axis grid lines.

distance (the distance from the true target to the nearest grid line) is constrained in $[-r/2, r/2]$. Therefore, we set the maximal adjustment range of all grid lines as $\pm r/2$. For example, for x-axis grid lines, $x_i \in [x_i^{(0)} - r/2, x_i^{(0)} + r/2]$, $i = 1, \ldots, n$, as can be seen in Figure 2.

Then, we can express the upper and lower bounds of the x-axis grid lines as

$$\mathbf{x}^{(u)} = \mathbf{x}^{(0)} + r/2 \cdot \mathbf{1} \tag{9}$$

$$\mathbf{x}^{(l)} = \mathbf{x}^{(0)} - r/2 \cdot \mathbf{1} \tag{10}$$

where $\mathbf{1}$ denotes the one vector. Similarly, we can obtain the upper and lower bounds of the y-axis grid lines as

$$\mathbf{y}^{(u)} = \mathbf{y}^{(0)} + r/2 \cdot \mathbf{1} \tag{11}$$

$$\mathbf{y}^{(l)} = \mathbf{y}^{(0)} - r/2 \cdot \mathbf{1} \tag{12}$$

Considering the bounds, problem (8) can be rewritten as

$$\begin{aligned}&\left[\tilde{\mathbf{w}}, \tilde{\boldsymbol{\Theta}}\right] = \arg\min_{\mathbf{w}, \boldsymbol{\Theta}} \left\{ \|\mathbf{z} - \mathbf{D}(\boldsymbol{\Theta})\mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_1 \right\} \\ &\text{s.t. } \boldsymbol{\Theta} = \mathbf{x} \times \mathbf{y} \\ &\mathbf{x}^{(l)} \le \mathbf{x} \le \mathbf{x}^{(u)} \\ &\mathbf{y}^{(l)} \le \mathbf{y} \le \mathbf{y}^{(u)}\end{aligned} \tag{13}$$

Furthermore, it is worth noting that the representation coefficient $\mathbf{w} = [w_1, w_2, \ldots, w_N]^{\mathrm{T}}$ encodes the number and locations of multiple targets. If a target is located on the ith grid points, then its element $w_i = 1$; otherwise, $w_i = 0$. Therefore, the representation coefficient w should be non-negative. Considering this, problem (13) can be given as

Algorithm 1 Dictionary refinement algorithm.

Input: $\mathbf{x}^{(0)}$, $\mathbf{y}^{(0)}$, $i = 0$, k_{out}, r_{th}
Iteration:
1: **while** $i<k_{out}$ and $r_i>r_{th}$ **do**
2: update $\tilde{\mathbf{w}}^{(i+1)}$ according to equation (15);
3: approximate the true $\mathbf{w}$ according to equation (16);
4: update $\tilde{\boldsymbol{\Theta}}^{(i+1)}$ according to equation (17);
5: calculate r_{i+1} according to equation (21);
6: $i \leftarrow i+1$.
7: **end while**
8: $\tilde{\mathbf{w}} \leftarrow \hat{\mathbf{w}}^{(i)}, \tilde{\Theta} \leftarrow \tilde{\boldsymbol{\Theta}}^{(i)}$.
Output:
$\tilde{\mathbf{w}}, \tilde{\boldsymbol{\Theta}}$

$$\begin{aligned} \left[\tilde{\mathbf{w}}, \tilde{\boldsymbol{\Theta}}\right] = & \arg\min_{\mathbf{w},\Theta}\left\{\|\mathbf{z}-\mathbf{D}(\boldsymbol{\Theta})\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_1\right\} \\ & \text{s.t. } \boldsymbol{\Theta} = \mathbf{x}\times\mathbf{y} \\ & \mathbf{x}^{(l)} \leq \mathbf{x} \leq \mathbf{x}^{(u)} \\ & \mathbf{y}^{(l)} \leq \mathbf{y} \leq \mathbf{y}^{(u)} \\ & 0 \leq \mathbf{w} \end{aligned} \tag{14}$$

where $\mathbf{0}$ denotes the zero matrix.

Localization via dictionary refinement

In this section, we first discuss how to solve problem (14) and then present how to achieve accurate localization using the solution.

Dictionary refinement algorithm

The process of the proposed dictionary refinement algorithm is summarized in Algorithm 1. The algorithm starts with an initial equi-spaced grid sampling $\boldsymbol{\Theta}^{(0)} = (\mathbf{x}^{(0)}, \mathbf{y}^{(0)})$, where $\mathbf{x}^{(0)}$ and $\mathbf{y}^{(0)}$ denote the initial values of x-axis and y-axis grid lines. Then, the grid $\boldsymbol{\Theta}$ is iteratively adjusted to reduce or even eliminate the dictionary mismatch.

We develop an iterative two-step algorithm to solve problem (14). The key idea of the algorithm is to alternately optimize over sparse signal $\mathbf{w}$ and parameter $\boldsymbol{\Theta}$ in each iteration. The first step of the algorithm optimizes over the representation coefficient $\mathbf{w}$ while keeping the grid parameters $\boldsymbol{\Theta}$ fixed

$$\begin{aligned} \tilde{\mathbf{w}}^{(i+1)} = & \arg\min_{\mathbf{w}}\left\{\left\|\mathbf{z} - D\left(\tilde{\boldsymbol{\Theta}}^{(i)}\right)\mathbf{w}\right\|_2^2 + \lambda\|\mathbf{w}\|_1\right\} \\ & \text{s.t. } \mathbf{0} \leq \mathbf{w} \end{aligned} \tag{15}$$

where superscripts denote algorithm iteration. This is a standard regularized LS optimization problem, and there exist many tools to solve this problem. In our implementation, we use the convex optimization software package.[24]

In practice, the solution $\tilde{\mathbf{w}}$ to equation (15) may only have a small number of significant coefficients and many negligible, but non-zero coefficients. These small coefficients contribute very little to the sparse signal. Therefore, we approximate the true w by neglecting the small coefficients

$$\hat{\mathbf{w}} = H(\tilde{\mathbf{w}}) : \hat{w}_i = \begin{cases} 0, & 20\log_{10}\left(\frac{\tilde{w}_i}{\max_j|\tilde{w}_j|}\right) < \delta \\ \tilde{w}_i, & \text{otherwise} \end{cases} \tag{16}$$

where δ is a small negative sparsity threshold.

The second step optimizes over the grid parameters $\boldsymbol{\Theta}$ while keeping the approximated representation coefficient $\hat{\mathbf{w}}$ fixed

$$\begin{aligned} \tilde{\boldsymbol{\Theta}}^{(i+1)} = & \arg\min_{\Theta}\left\{\left\|z - \mathbf{D}(\boldsymbol{\Theta})\hat{\mathbf{w}}^{(i+1)}\right\|_2^2\right\} \\ & \text{s.t. } \boldsymbol{\Theta} = \mathbf{x}\times\mathbf{y} \\ & \mathbf{x}^{(l)} \leq \mathbf{x} \leq \mathbf{x}^{(u)} \\ & \mathbf{y}^{(l)} \leq \mathbf{y} \leq \mathbf{y}^{(u)} \end{aligned} \tag{17}$$

We solve the problem using a trust-region subspace method[25] where the gradient of objective function must be provided. For the sake of convenience, we transform the grid parameters $\boldsymbol{\Theta}$ into a vector $\boldsymbol{\theta} = [\mathbf{x}^{\mathrm{T}}, \mathbf{y}^{\mathrm{T}}]^{\mathrm{T}} = [\theta_1, \ldots, \theta_{2n}]^{\mathrm{T}}$ (notably, a grid is determined by grid lines or grid points; therefore, we will use $\boldsymbol{\theta}$ and $\boldsymbol{\Theta}$ interchangeably to represent an example of grid in the rest of this article). Next, we deduce the gradient of objective function with respect to $\boldsymbol{\theta}$.

Let

$$F(\boldsymbol{\Theta}) = \|\mathbf{z} - \mathbf{D}(\boldsymbol{\Theta})\mathbf{w}\|_2^2 \tag{18}$$

Denote by $\nabla_i F(\boldsymbol{\Theta})$, the derivation of $F(\boldsymbol{\Theta})$ with respect to θ_i, $1 \leq i \leq 2n$, then we have

$$\nabla_i F(\boldsymbol{\Theta}) = -2(\mathbf{z} - \mathbf{D}\mathbf{w})^{\mathrm{T}}(\nabla_i \mathbf{D})\mathbf{w} \tag{19}$$

where $\nabla_i \mathbf{D}$ indicates the gradient of the sparsifying dictionary $\mathbf{D}$ with respect to θ_i, where $1 \leq i \leq 2n$. As a matter of fact, according to the grid numbering rule, only n atoms of $\mathbf{D}$ are dependent on θ_i while the others are not. Therefore, the gradient of F can be given by (20)

$$\nabla_i F(\boldsymbol{\Theta}) = \begin{cases} -2(\mathbf{z}-\mathbf{D}\mathbf{w})^{\mathrm{T}} \sum\limits_{j = i:n:n(n-1)+i} w_j \nabla_i \mathbf{d}_j & 1 \leqslant i \leqslant n \\ -2(\mathbf{z}-\mathbf{D}\mathbf{w})^{\mathrm{T}} \sum\limits_{j = n(i-n-1)+1:n(i-n)} w_j \nabla_i \mathbf{d}_j & n+1 \leqslant i \leqslant 2n \end{cases} \tag{20}$$

After each iteration, the current system cost is calculated as

$$r_{i+1} = \left\| \mathbf{z} - \mathbf{D}\left(\tilde{\boldsymbol{\Theta}}^{(i+1)}\right)\tilde{\mathbf{w}}^{(i+1)} \right\|_2^2 + \lambda \left\| \tilde{\mathbf{w}}^{(i+1)} \right\|_1 \quad (21)$$

Remark 1. Algorithm 1 is a two-layer iteration procedure. In each outer iteration, sparse signal $\mathbf{w}$ and parameter $\boldsymbol{\Theta}$ are updated sequentially with maximum outer iteration number k_{out}. Besides, the optimization of parameter $\boldsymbol{\Theta}$ is also an iteration procedure with maximum inner iteration number k_{in}.

Remark 2. The algorithm stops when the convergence condition is reached, for example, when the iteration has been executed the maximum outer iteration number k_{out} or the current system cost is smaller than the cost threshold r_{th}.

Localization

The output results $\tilde{\mathbf{w}}$ and $\tilde{\boldsymbol{\Theta}}$ encode the number and locations of considered targets. The estimated target number $\hat{K}$ and target locations $\hat{\boldsymbol{\theta}}_k$ are given by

$$\hat{K} = \|\hat{\mathbf{w}}\|_0 \quad (22)$$

$$\hat{\boldsymbol{\theta}}_k = \tilde{\boldsymbol{\theta}}_{I_k} \quad (23)$$

where $\{I_k\}$ is an ordered set such that $|\hat{w}_{I_k}|$ is the kth-largest element of $\hat{\mathbf{w}}$.

Numerical evaluation

In this section, we conduct simulations to evaluate the effectiveness and robustness of the proposed dictionary refinement algorithm.

Simulation setup

All simulations are conducted on MATLAB R2015b. We consider a $9\text{m} \times 9\text{m}$ square area with $K = 3$ targets. To apply CS, we sample the area into a 10×10 grid, that is, $r = 1\text{m}$. RSS measurements from these targets are collected by $M = 36$ sensors. Unless otherwise stated, we use the same scenario in the later simulations. For the sake of convenience, we refer to the proposed dictionary refinement algorithm as DicRef and compare it with the traditional CS reconstruction algorithms, including BP, OMP, and SBL.

In order to assign an estimated location to a target, we compute all pairs of the distances between $\boldsymbol{\tau}_k$ and $\hat{\boldsymbol{\theta}}_k$ and sort them in a non-decreasing order. Based on the sorted list, we assign a target to the first unused estimated location.

Definition 1. The average localization error, denoted by $Avg.Error$, is defined to be the ratio of the average distance between the estimated and actual locations of all targets versus the minimum between the estimated and actual target numbers

$$Avg.Error = \frac{\sum_{k=1}^{K_{min}} \left\| \boldsymbol{\tau}_k - \hat{\boldsymbol{\theta}}_k \right\|_2}{K_{min}} \quad (24)$$

where $\boldsymbol{\tau}_k$ and $\hat{\boldsymbol{\theta}}_k$ denote the true and estimated locations for the kth target, respectively, and $K_{min} = \min\{K, \hat{K}\}$. All results are averaged over 100 random trials.

We have observed from simulations that the decrement of objective function usually becomes quite small within very few iterations for trust-region subspace method. Based on the observation, to reduce time cost, we set the maximum inner iteration number $k_{in} = 3$ in our simulations. Table 1 summarizes some parameter values used for our simulations.

Table 1. Parameter values for simulations.

Parameters	Explanation	Values
λ	Regularization parameter	1
δ	Sparsity threshold	−5 dB
d_0	Reference distance	1 m
γ	Path loss coefficient	2
r_{th}	Cost threshold	0.01
k_{out}	Maximum iteration number	100

Simulation without off-grid targets

We first consider an ideal situation where all targets are located exactly on the initial grid. The true locations of three targets are set to (4, 3), (8, 6), and (5, 8). Figure 3 shows the counting and localization results of different algorithms. It can be seen that all algorithms accurately estimate the number of these targets. In terms of localization, it can be seen that BP, SBL, and DicRef are able to accurately estimate the locations of multiple targets. In contrast, OMP fails to localize the first target.

Simulation with off-grid targets

Then, we consider a more general situation where there exist targets deviating from the initial grid. The true positions of three targets are set to (3.85, 3.47), (8.42, 5.84), and (5.35, 8.26), respectively. In such case, all targets are located off the initial grid. Figure 4 shows the counting and localization results of different algorithms.

As can be shown in Figure 4, when there is no noise, BP, SBL, and DicRef correctly estimate the number of targets, while OMP suffers from false targets. As for localization, the traditional CS reconstruction algorithms localize three targets to the nearest grid points. Unsurprisingly, these behaviors are the natural results

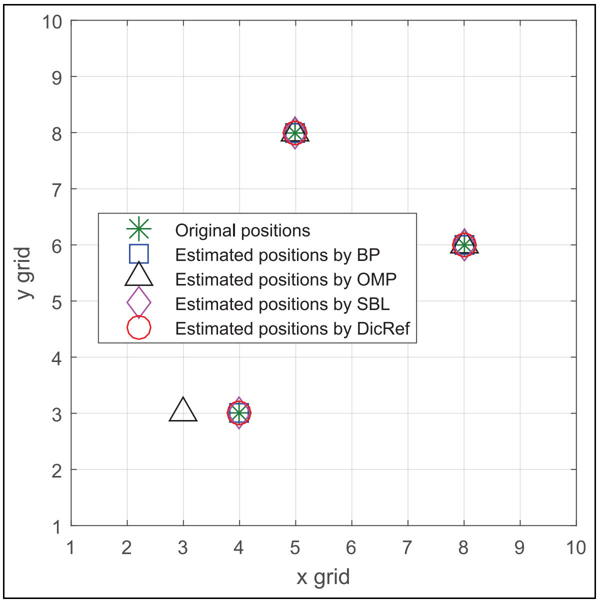

Figure 3. Counting and localization results of different algorithms when all targets fall exactly on the initial grid.

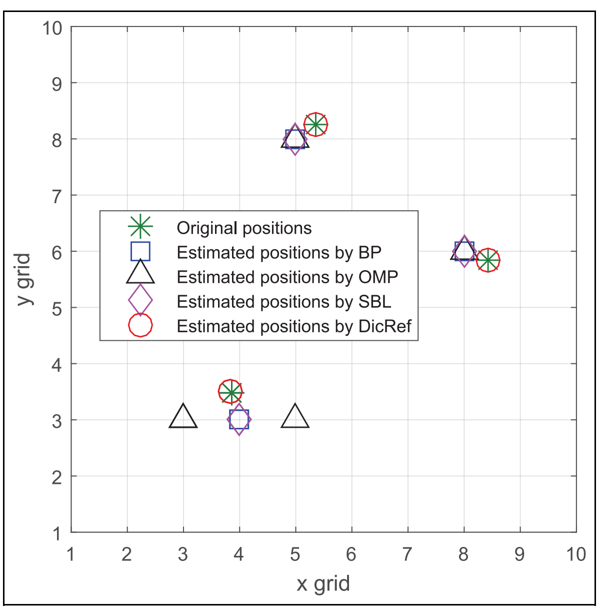

Figure 4. Counting and localization results of different algorithms when there exist targets deviating from the initial grid.

incurred by dictionary mismatch. In contrast, the positions estimated by DicRef are so accurate that they are visually indistinguishable from their original positions. This example clearly indicates that, compared with the traditional CS reconstruction algorithms, the proposed dictionary refinement algorithm is able to provide better performance when there are targets deviating from the initial grid.

Effect of measurement noise

It is inevitable for measurements to be corrupted with environmental noise. In order to check the robustness of the proposed dictionary refinement algorithm against measurement noise, we intentionally add Gaussian white noise $\mathcal{N}(0, \sigma^2\mathbf{I})$ to measurements, where σ^2 denotes the variance of noise. We define the signal-to-noise ratio (SNR) as $10\log_{10}(\|\mathbf{z}\|_2^2/(M\sigma^2))$.

Figure 5 reports the average estimated target numbers, probabilities of correct counting, and average localization errors of different algorithms under different measurement noise levels. As can be observed, with the increasing of SNR, the average estimated target numbers of all algorithms gradually approximate to the true target number, which is indicated by the horizontal baseline in Figure 5(a). When SNR is low, the proposed dictionary refinement algorithm performs much better than the other algorithms. As SNR is high, BP, SBL, and DicRef can estimate the number of targets accurately while OMP suffers from false targets. As for the probability of correct counting, the behaviors of different algorithms are similar with the average estimated target numbers. The probabilities of correct counting with respect to all algorithms increase with the increasing of SNR. When noise is serious, DicRef performs better than the other algorithms. As noise is slight, the probabilities of correct counting with respect to BP, SBL, and DicRef approximate to 1. However, the probability of correct counting of OMP keeps below 0.21 in the entire SNR range. Additionally, the average localization errors for all algorithms decrease as the SNR increases. As a matter of fact, this is reasonable as the signal reconstruction accuracy is proportional to SNR. Another important observation is that the proposed dictionary refinement algorithm performs better than the other algorithms no matter how high the noise level is. The reason lies in the fact that the proposed algorithm dynamically adjusts the grid to refine the dictionary while the other algorithms do not. Furthermore, we can see that the proposed dictionary refinement algorithm can tolerate a certain level of measurement noise. For example, when SNR is higher than 30 dB, the average localization error with respect to DicRef is no more than 0.1 m.

Then, to further examine the performance of the proposed dictionary refinement algorithm, we compare the cumulative distribution functions of the average localization errors of different algorithms when SNR = 20 dB. The results are shown in Figure 6. An important observation is that the proposed dictionary refinement algorithm performs significantly better than the traditional CS reconstruction algorithms. For example, when the average localization error is limited under 1 m, the proposed dictionary refinement algorithm DicRef achieves an impressive improvement of 15% over SBL, 45% over BP, and 57% over OMP.

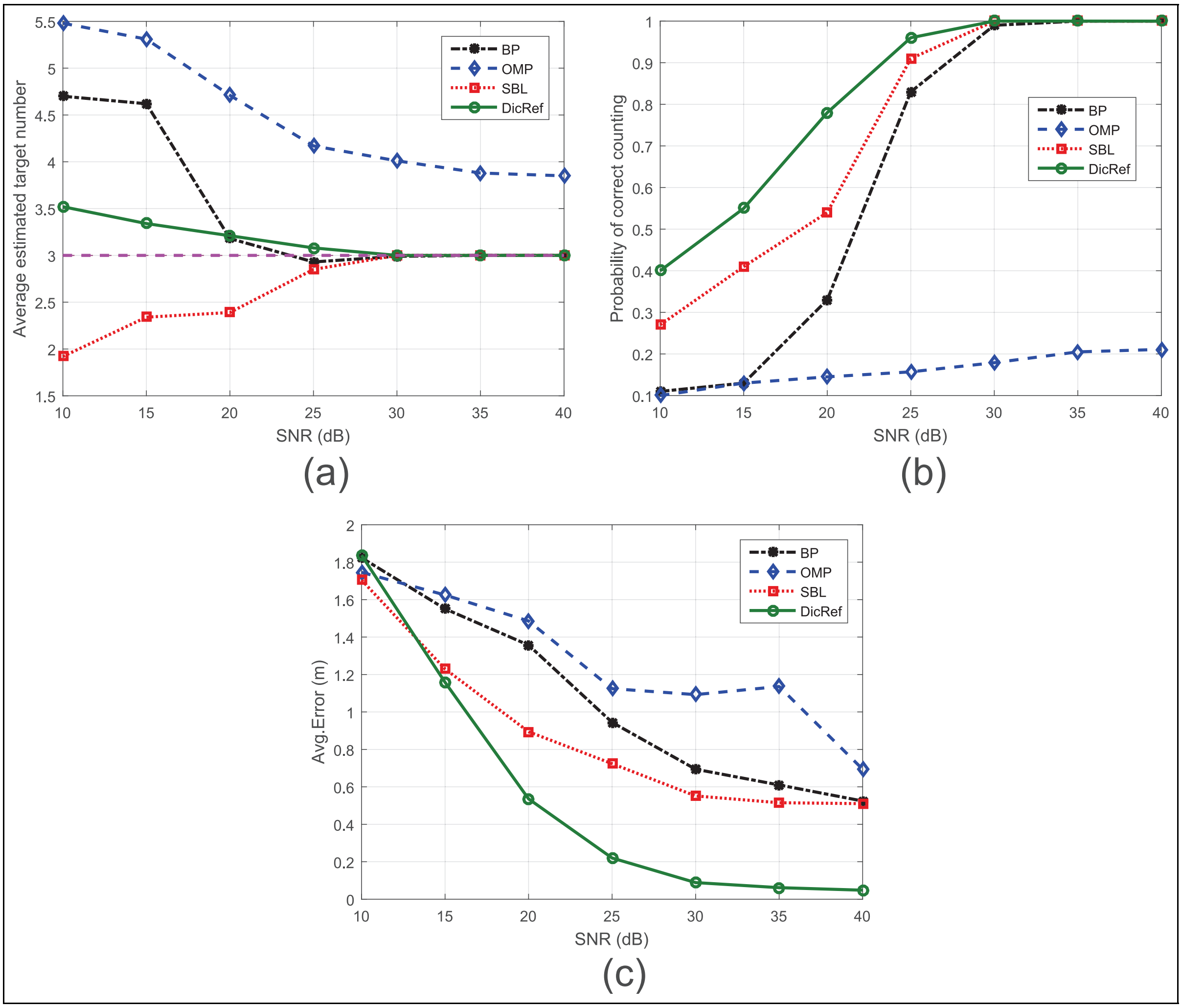

Figure 5. Counting and localization performance versus SNR when measurement number $M = 36$: (a) average estimated target number, (b) probability of correct counting, and (c) conditional average localization error.

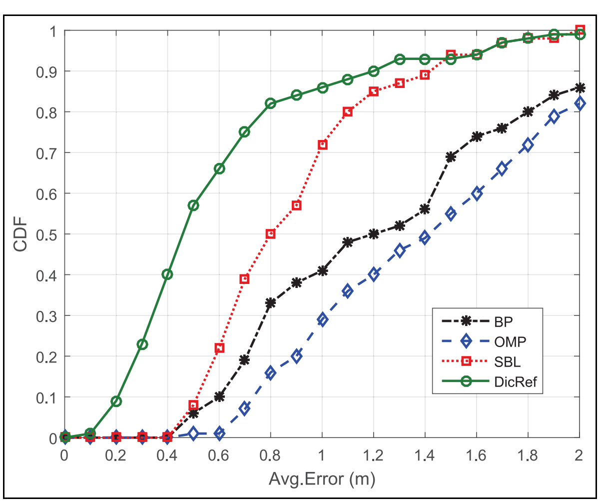

Figure 6. Cumulative distribution functions (CDFs) of average location errors.

Effect of measurement number

Now, we check the performance of the proposed dictionary refinement algorithm with different measurement numbers. We set target number $K = 5$, SNR $= 20$ dB and vary measurement number M from 30 to 36. The results are plotted in Figure 7, where the horizontal baseline indicates the true target number. It can be seen that, as expected, increasing measurement number brings about both counting and localization performance improvement for all algorithms. It is reasonable because the sparse recovery accuracy is proportional to measurement number when the other parameters are fixed. More importantly, compared to the others, the proposed algorithm DicRef shows significantly superior performance. The superiority can be attributed to the fact that the proposed algorithm dynamically adjusts the grid to refine the dictionary while the others suffer from serious dictionary mismatch.

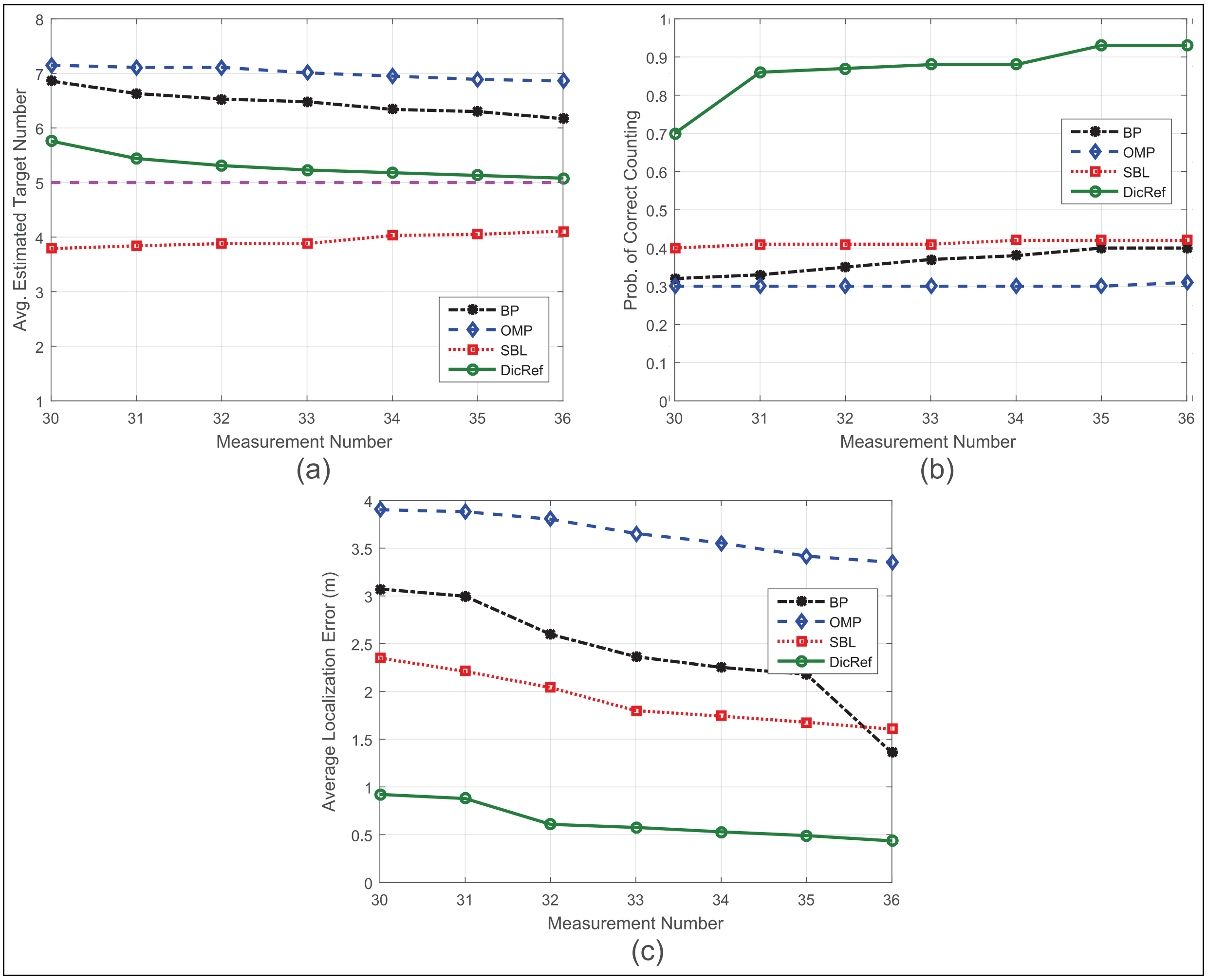

Figure 7. Counting and localization performance versus measurement number when SNR = 20 dB: (a) average estimated target number, (b) probability of correct counting, and (c) conditional average localization error.

Results on different parameters

Then, to further test the performance of the proposed algorithm, we sample the same area into a denser grid with 19×19 grid points. In such a case, the signal length $N = 361$, and the parameter $r = 0.5m$. We set target number $K = 5$ and measurement number $M = 100$, the counting and localization results are shown in Figure 7. From Figure 7, the traditional CS reconstruction algorithms miss most targets and only estimate few targets on the near grid points. Obviously, the inferior performance is caused by dictionary mismatch. Contrastively, the proposed dictionary refinement algorithm accurately localizes all targets, and the errors are negligible. The superior performance attributes to the fact that DicRef is able to dynamically adjust grid to refine the assumed dictionary.

Complexity analysis

Eventually, we analyze the complexities of different algorithms in terms of computational complexity and CPU running times. The results are shown in Table 2. For BP, OMP, and SBL, the computational complexity are $O(N^3)$, $O(NK^2)$, and $O(NM^2)$, respectively. As for DicRef, the computational complexity is dominated by the optimization of sparse signal $\mathbf{w}$ and parameter $\boldsymbol{\Theta}$. The former is $O(N^3)$ and the latter is reduced by restricting the inner iteration number to 3. In terms of average CPU running times (over 100 trials), it can be

Table 2. Complexity analysis of different algorithms.

Algorithms	Average error (m)	Complexity	CPU running time (s)
BP	1.39	$O(N^3)$	0.17
OMP	1.51	$O(NK^2)$	0.01
SBL	0.91	$O(NM^2)$	0.14
DicRef	0.56	$O(N^3)$	23.74

BP: basis pursuit; OMP: orthogonal matching pursuit; SBL: sparse Bayesian learning.

clearly seen that DicRef is much more accurate than the other algorithms while it is inferior to the other algorithms in terms of CPU running time. The inferiority is mainly attributed to the following two factors. On one hand, DicRef is an iterative algorithm and it usually converges until a number of iterations. On the other hand, in practice, we observe that the grid parameter optimization is time consuming in each iteration. As a matter of fact, the running time can be reduced with slight accuracy loss by setting the maximum outer iteration number k_{out} to be a smaller value. To sum up, we can conclude that the proposed localization method can be applied in the scenarios which are sensitive to accuracy but insensitive to real time.

Conclusion

In this article, we investigated the dictionary mismatch problem in CS-based localization and developed an efficient dictionary refinement algorithm. Different from the other CS reconstruction algorithms using a fixed grid, the proposed dictionary refinement algorithm dynamically adjusts the grid to alleviate or even eliminate dictionary mismatch. To achieve this, we view the sparsifying dictionary as a parameterized dictionary, with the sampled grid as adjustable parameters; then the localization problem is transformed into a joint sparse signal recovery and parameter estimation problem; at last, an iterative two-step algorithm is developed to solve the joint optimization problem. Simulation results show that, compared to the most existing CS reconstruction algorithms, the proposed dictionary refinement algorithm achieves better performance with high time cost. In the future, we will seek for alternative solutions to reduce the cost of the proposed algorithm at the expense of allowed performance loss.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (Grant Nos 61571463, 61472445 Natural Science Foundation of Jiangsu Province, China (Grant No BK20171401)).

References

1. Akyildiz IF, Su W, Sankarasubramaniam Y, et al. A survey on sensor networks. *IEEE Comm Mag* 2002; 40(8): 102–114.
2. US Coast Guard Navigation Center. *Global positioning system standard positioning service specification*. Alexandria, VA: US Coast Guard Navigation Center, 1995.
3. Donoho DL. Compressed sensing. *IEEE Trans Inform Theor* 2006; 52(4): 1289–1306.
4. Baraniuk RG. Compressive sensing. *IEEE Signal Process Mag* 2007; 24(4): 1–9.
5. Herman MA and Strohmer T. General deviants: an analysis of perturbations in compressed sensing. *IEEE J Sel Top signa* 2010; 4(2): 342–349.
6. Chi Y, Scharf LL and Pezeshki A. Sensitivity to basis mismatch in compressed sensing. *IEEE Trans Signal Process* 2011; 59(5): 2182–2195.
7. Cevher V, Duarte MF and Baraniuk RG. Distributed target localization via spatial sparsity. In: *Proceedings of the 2008 IEEE 16th European signal processing conference*, Lausanne, 25–29 August 2008, pp.1–5. New York: IEEE.
8. Cevher V, Boufounos P, Baraniuk RG, et al. Near-optimal bayesian localization via incoherence and sparsity. In: *Proceedings of the 2009 international conference on information processing in sensor networks (IPSN)*, San Francisco, CA, 13–16 April 2009, pp.205–216. New York: IEEE.
9. Feng C, Valaee S and Tan Z. Multiple target localization using compressive sensing. In: *Proceedings of the global telecommunications conference (GLOBECOM)*, Honolulu, HI, 30 November–4 December 2009, pp.1–6. New York: IEEE.
10. Feng C, Au WSA, Valaee S, et al. Compressive sensing based positioning using RSS of WLAN access points. In: *Proceedings of the 29th conference on Information communications*, San Diego, CA, 14–19 March 2010, pp.1–9. Piscataway, NJ: IEEE Press.
11. Feng C, Au WSA, Valaee S, et al. Received-signal-strength-based indoor positioning using compressive sensing. *IEEE Trans Mobile Comput* 2012; 11(12): 1983–1993.
12. Zhang B, Cheng X, Zhang N, et al. Sparse target counting and localization in sensor networks based on compressive sensing. In: *Proceedings of the INFOCOM*, Shanghai, China, 10–15 April 2011, pp.2255–2263. New York: IEEE.
13. Au AWS, Feng C, Valaee S, et al. Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device. *IEEE Trans Mobile Comput* 2013; 12(10): 2050–2062.
14. Guyen GK, Van Nguyen T and Shin H. Learning dictionary and compressive sensing for WLAN localization. In: *Proceedings of the 2014 IEEE wireless communications and networking conference (WCNC)*, Istanbul, Turkey, 6–9 April 2014, pp.2910–2915. New York: IEEE.
15. Zhang Y, Zhu Y, Lu M, et al. Using compressive sensing to reduce fingerprint collection for indoor localization. In: *Proceedings of the 2013 wireless communications and networking conference (WCNC)*, Shanghai, China, 7–10 April 2013, pp.4540–4545. New York: IEEE.
16. Liu L, Cui T and Lv W. A range-free multiple target localization algorithm using compressive sensing theory in wireless sensor networks. In: *Proceedings of the IEEE 11th international conference on mobile ad hoc and sensor*

systems (MASS), Philadelphia, PA, 28–30 October 2014, pp.690–695. New York: IEEE.
17. Nguyen TLN and Shin Y. Multiple target localization in WSNs based on compressive sensing using deterministic sensing matrices. *Int J Distrib Sens Netw* 2015; 2015: 1–9.
18. Sun B, Guo Y, Li N, et al. TDL: two-dimensional localization for mobile targets using compressive sensing in wireless sensor networks. *Comput Comm* 2016; 78: 45–55.
19. Candes EJ. The restricted isometry property and its implications for compressed sensing. *Compt Rendus Math* 2008; 346(9): 589–592.
20. Chen S, Donoho D and Saunders M. Atomic decomposition by basis pursuit. *SIAM J Sci Comput* 1998; 20(1): 33–61.
21. Tropp JA and Gilbert AC. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans Inform Theor* 2007; 53(12): 4655–4666.
22. Tipping ME. Sparse Bayesian learning and the relevance vector machine. *J Mach Learn Res* 2001; 1: 211–244.
23. Clouqueur T, Saluja KK and Ramanathan P. Fault tolerance in collaborative sensor networks for target detection. *IEEE Trans Comput* 2004; 53(3): 320–333.
24. Grant M and Boyd S. CVX: MATLAB software for disciplined convex programming (version 1.21), 2011, http://cvxr.com/cvx/
25. Branch MA, Coleman TF and Li Y. A subspace, interior, and conjugate gradientmethod for large-scale bound-constrained minimization problems. *SIAM J Sci Comput* 1999; 21: 1–23.