




Article

Sparsity-Based Two-Dimensional DOA Estimation for Co-Prime Planar Array via Enhanced Matrix Completion

Donghe Liu, Yongbo Zhao *  and Tingxiao Zhang

National Lab of Radar Signal Processing, Xidian University, Xi'an 710071, China

* Correspondence: ybzhao@xidian.edu.cn

Abstract: In this paper, the two-dimensional (2-D) direction-of-arrival (DOA) estimation problem is explored for the sum-difference co-array (SDCA) generated by the virtual aperture expansion of co-prime planar arrays (CPPA). Since the SDCA has holes, this usually causes the maximum virtual aperture of CPPA to be unavailable. To address this issue, we propose a complex-valued, sparse matrix recovery-based 2-D DOA estimation algorithm for CPPA via enhanced matrix completion. First, we extract the difference co-arrays (DCA) from SDCA and construct the co-array interpolation model via nuclear norm minimization to initialize the virtual uniform rectangular array (URA) that does not contain the entire rows and columns of holes. Then, we utilize the shift-invariance structure of the virtual URA to construct the enhanced matrix with a two-fold Hankel structure to fill the remaining empty elements. More importantly, we apply the alternating direction method of the multipliers (ADMM) framework to solve the enhanced matrix completion model. To reduce the computational complexity of the traditional vector-form, sparse recovery algorithm caused by the Kronecker product operation between dictionary matrices, we derive a complex-valued sparse matrix-recovery model based on the fast iterative shrinkage-thresholding (FISTA) method. Finally, simulation results demonstrate the effectiveness of the proposed algorithm.



Citation: Liu, D.; Zhao, Y.; Zhang, T. Sparsity-Based Two-Dimensional DOA Estimation for Co-Prime Planar Array via Enhanced Matrix Completion. *Remote Sens.* **2022**, *14*, 4690. <https://doi.org/10.3390/rs14194690>

Academic Editor: Gwanggil Jeon

Received: 26 August 2022

Accepted: 15 September 2022

Published: 20 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: co-prime planar array; enhanced matrix completion; sparse matrix recovery; two-dimensional direction-of-arrival estimation

1. Introduction

Direction-of-arrival (DOA) estimation is a fundamental issue for array signal processing and has been widely applied in radar, sonar, and navigation [1–3]. Various methods have been proposed for DOA estimation, such as classical subspace-based MUSIC and ESPRIT methods [4–9]. However, these methods rely on uniform linear arrays (ULA) or uniform rectangular arrays (URA), leading to mutual coupling and redundancy problems between sensors. Recently, a novel coprime array (CPA) [10,11] has been proposed and received extensive attention. Compared with the ULA, the CPA has a virtual aperture expansion capability by calculating its difference co-array (DCA), improving angle estimation performance. In addition, the distance between the sensors of the CPA is no longer limited to the half-wavelength, which can alleviate the mutual coupling effect between the physical sensors.

In order to exploit the virtual aperture of the CPA, many DOA estimation algorithms have been proposed in [12–18]. A spatial smoothing-based MUSIC algorithm [12] and a sparse recovery-based method [13] have been proposed which utilize the consecutive parts of the DCA and discard the non-consecutive parts for DOA estimation. However, none of these methods utilize the non-consecutive part of the DCA, which usually causes the virtual aperture loss of the DCA, reducing the DOA estimation performance. In order to make full use of the virtual aperture of the CPA, co-array interpolation-based algorithms [14,15] were proposed to fill the holes in the DCA. In [16], the holes in the DCA are filled by minimizing the nuclear norm of the covariance matrix of the received

signal, generating a consecutive virtual ULA to improve angle estimation performance. In [17,18], the covariance matrix with the Toeplitz structure corresponding to the interpolated virtual array was reconstructed through atomic norm minimization, improving the DOA estimation accuracy. In [19,20], co-array interpolation algorithms based on truncated nuclear norm regularization were proposed to interpolate the covariance matrix with the Toeplitz structure corresponding to a virtual ULA, thereby improving the virtual aperture of the CPA. A 2-D DOA estimation algorithm was proposed for coprime parallel arrays via sparse representation in [21], extending the advantages of CPA. Although this algorithm implies sparse representation to improve the performance of DOA estimation, it does not utilize the maximum virtual aperture of the coprime parallel array, so there is an inherent aperture loss. In [22], a 2-D DOA fast estimation of multiple signals with matrix completion theory in a coprime planar array was proposed. This algorithm uses the matrix completion technique to complete the part holes of the virtual URA, which limits the number of DOFs. In addition, the missing elements in the entire rows and columns of the virtual URA are not accurately filled, which reduces recovery performance. In [23], a partial spectral search algorithm was proposed for the coprime planar array (CPPA), which improves DOA estimation accuracy by exploiting the coprime property to eliminate ambiguity. To benefit from the virtual aperture expansion of CPPA, Shi et al. in [24] proposed a sparsity-based, two-dimensional (2-D) DOA estimation algorithm for the sum-difference co-array (SDCA) of CPPA. However, this algorithm only utilizes the consecutive parts of the SDCA and discards the non-consecutive parts, resulting in an inherent loss of virtual aperture. Furthermore, co-array interpolation-based methods [14–17] cannot be used to fill the holes in SDCA. The reason is that the holes of SDCA have empty elements of the entire rows and columns, so the nuclear norm minimization-based algorithm [25] fails to recover the missing elements.

This paper proposes a sparse matrix-recovery-based 2-D DOA estimation algorithm via enhanced matrix completion to fill the holes in the SDCA. Using the coprime property, we first establish the SDCA of CPPA by rearranging the cross-correlation matrix of two coprime planar sub-arrays. Since the SDCA has missing elements in the entire rows and columns, the nuclear norm minimization-based matrix completion algorithm cannot be utilized to fill the holes. To solve this problem, we interpolate the covariance matrices of the DCA with the Toeplitz structure, thereby initializing the SDCA that does not contain the entire rows or columns of empty elements. Then, we use the shift-invariance structure of the SDCA to construct the enhanced matrix with a two-fold Hankel structure to fill all the empty elements accurately. Subsequently, we construct the nuclear norm, minimization-based, enhanced matrix completion model. Then, we utilize the multi-block, alternating direction method of the multipliers (ADMM) [26] framework for solving this model to reduce the computational burden caused by the increase in enhanced matrix dimensions. More importantly, the proposed enhanced matrix completion model can obtain the maximum virtual aperture of the CPPA, improving the angle estimation accuracy. Finally, we also derive a complex-valued, sparse matrix-recovery model based on the fast iterative shrinkage-thresholding (FISTA) method [27] reducing the computational complexity of the traditional vector-form, sparse recovery algorithm.

The main contributions of the proposed algorithm are as follows.

- We extract the DCA from the SDCA and use the co-array interpolation algorithm to fill the holes in the DCA, thereby constructing a virtual URA model that does not contain the missing elements of the entire rows and columns.
- To accurately complete all the empty elements of the virtual URA, we construct the enhanced matrix-completion model. Furthermore, we derive an enhanced matrix-completion algorithm based on the ADMM framework, reducing the computational complexity of the matrix completion.
- We derive a complex-valued, sparse matrix-recovery algorithm based on the FISTA method, which avoids the Kronecker product operation between dictionary matrices, reducing the computational complexity of the traditional vector-form, sparse recovery algorithm.

The rest of this paper is organized as follows. In Section 2, we present the coprime planar array signal model, establish an enhanced matrix-completion model, and derive a matrix-completion algorithm based on the ADMM framework. In addition, we derive a complex-valued, sparse matrix-recovery algorithm based on the FISTA method. In Section 3, we demonstrate the simulation results. Discussion and conclusions are provided in Section 4 and Section 5, respectively.

Notations: We use lowercase bold letters for vectors, uppercase bold letters for matrices, and italics for scalars. $(\cdot)^T$, $(\cdot)^*$ and $(\cdot)^H$ represent transpose, complex conjugate and conjugate transpose, respectively. $(\cdot)^{-1}$ is matrix inversion. \otimes and \circ represent the Kronecker product and Hadamard product, respectively. $\text{diag}(\cdot)$ and $\text{vec}(\cdot)$ are the diagonalization and vectorization operations. \mathbf{I}_N denotes the $N \times N$ identity matrix, and \mathbf{J}_N represents the $N \times N$ exchange matrix, with ones on its anti-diagonal and zeros elsewhere. $\nabla(\cdot)$ denotes the conjugate gradient of a complex-valued vector or matrix. $\langle \mathbf{A}, \mathbf{B} \rangle$ denotes the inner product of \mathbf{A} and \mathbf{B} . $\text{tr}(\cdot)$ refers to the trace of the matrix. The subscript $(\cdot)_+$ refers to a scalar $u_+ > 0$, then $u_+ = u$ and is equal to zero otherwise. $\|\cdot\|_1$, $\|\cdot\|_2$, and $\|\cdot\|_F$ are the l_1 -norm, l_2 -norm, and Frobenius norm, respectively.

2. Method

In this section, we first describe the signal model for CPPA. Then, we define the concept of SDCA by extracting the cross-correlation matrix of CPPA. Since there are entire rows and columns of missing elements in the SDCA, two co-array interpolation optimization problems are proposed to initialize the holes in the DCA to generate a virtual URA that does not contain the entire rows and columns of empty elements. Subsequently, we construct a nuclear norm, minimization-based, enhanced matrix-completion model to recover all missing elements in the virtual URA. Finally, a complex-valued, sparse matrix-recovery algorithm based on the FISTA method is derived to estimate the 2-D DOA of targets.

2.1. Signal Model

A CPPA consists of two planar sub-arrays with $2M \times 2M$ and $N \times N$ physical sensors, respectively, where M and N are coprime integers. As shown in Figure 1, the CPPA contains a total of $T = 4M^2 + N^2 - 1$ physical sensors, and the sensor position sets of the first and second planar sub-array are \mathbb{P}_1 and \mathbb{P}_2 , respectively. Therefore, the sensor position set of the CPPA is \mathbb{P} and can be expressed as

$$\begin{aligned} \mathbb{P} &= \mathbb{P}_1 \cup \mathbb{P}_2 \\ &= \{(m_x d_1, m_y d_1)\} \cup \{(n_x d_2, n_y d_2)\} \end{aligned} \quad (1)$$

where $0 \leq m_x, m_y \leq 2M - 1$, $0 \leq n_x, n_y \leq N - 1$, $d_1 = Nd$ and $d_2 = Md$ are the inter-element spacing of the first sub-array and the second sub-array along the x and y directions, respectively, and $d = \lambda/2$, λ is the signal wavelength.

Assume there are K uncorrelated narrowband far-field signals impinging the CPPA from direction $(\boldsymbol{\theta}, \boldsymbol{\varphi}) = [(\theta_1, \varphi_1), (\theta_2, \varphi_2), \dots, (\theta_K, \varphi_K)]$, where θ_k and φ_k denote the elevation and azimuth of the k th signal. We present the main assumptions of the signal model in Table 1.

Table 1. The main assumptions of the signal model.

The structure of array	$M > 1$ and $N > 1$ are co-prime integers
The statistical properties of the signal	follow the independent Gaussian distribution
The statistical properties of the noise	complex-valued additive white Gaussian noise
The inter-element spacing of the first sub-array	$d_1 = Nd$
The inter-element spacing of the second sub-array	$d_2 = Md$
The transform frequency domain variable u	$\sin(\theta) \cos(\varphi)$
The transform frequency domain variable v	$\sin(\theta) \sin(\varphi)$

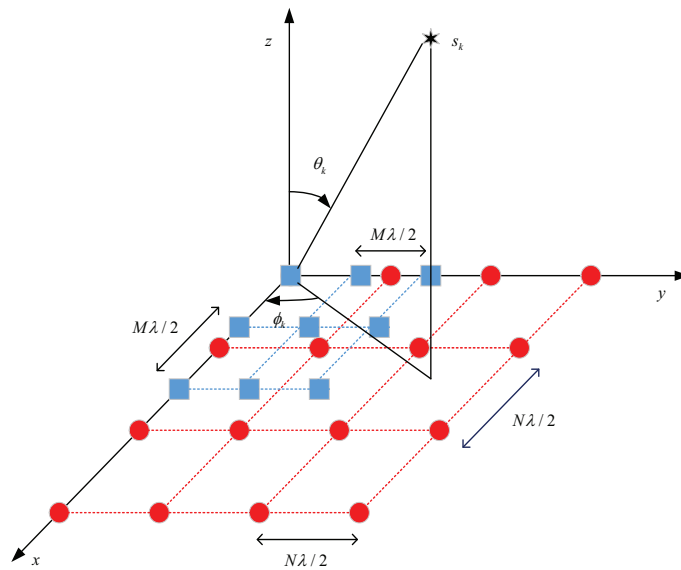


Figure 1. The configuration of the co-prime planar array with $M = 2$ and $N = 3$, where the red elements represent the array elements of the first planar sub-array, and the blue elements represent the array elements of the second planar sub-array.

After vectorization, the received signals of the first planar sub-array in the vector form of the q th snapshot can be expressed as

$$\mathbf{x}_1(q) = \mathbf{A}\mathbf{s}(q) + \mathbf{n}_1(q) \tag{2}$$

where $\mathbf{s}(q) = [s_1(q), s_2(q), \dots, s_K(q)]^T$ is the complex-valued signal vector, $\mathbf{n}_1(q)$ is the complex-valued white Gaussian noise with zero mean and variance σ^2 , $\mathbf{A} = \mathbf{A}_x \otimes \mathbf{A}_y$, $\mathbf{A}_x = [\mathbf{a}_x(u_1), \mathbf{a}_x(u_2), \dots, \mathbf{a}_x(u_K)]$, $\mathbf{A}_y = [\mathbf{a}_y(v_1), \mathbf{a}_y(v_2), \dots, \mathbf{a}_y(v_K)]$. $\mathbf{a}_x(u_k)$ and $\mathbf{a}_y(v_k)$ are the steering vectors along the x and y directions, respectively, and can be expressed as

$$\mathbf{a}_x(u_k) = [1, e^{-j\pi Nu_k}, \dots, e^{-j\pi(2M-1)Nu_k}]^T \tag{3}$$

$$\mathbf{a}_y(v_k) = [1, e^{-j\pi Nv_k}, \dots, e^{-j\pi(2M-1)Nv_k}]^T \tag{4}$$

where $u_k = \sin(\theta_k) \cos(\varphi_k)$ and $v_k = \sin(\theta_k) \sin(\varphi_k)$ represent the transform frequency domain variables. Similarly, the received signal model of the second planar sub-array can be given as

$$\mathbf{x}_2(q) = \mathbf{B}\mathbf{s}(q) + \mathbf{n}_2(q) \tag{5}$$

where $\mathbf{n}_2(q)$ is the complex-valued, white Gaussian noise vector, $\mathbf{B} = \mathbf{B}_x \otimes \mathbf{B}_y$, $\mathbf{B}_x = [\mathbf{b}_x(u_1), \mathbf{b}_x(u_2), \dots, \mathbf{b}_x(u_K)]$, $\mathbf{B}_y = [\mathbf{b}_y(v_1), \mathbf{b}_y(v_2), \dots, \mathbf{b}_y(v_K)]$. $\mathbf{b}_x(u_k)$ and $\mathbf{b}_y(v_k)$ can be represented as

$$\mathbf{b}_x(u_k) = [1, e^{-j\pi Mu_k}, \dots, e^{-j\pi(N-1)Mu_k}]^T \tag{6}$$

$$\mathbf{b}_y(v_k) = [1, e^{-j\pi Mv_k}, \dots, e^{-j\pi(N-1)Mv_k}]^T \tag{7}$$

2.2. The Virtual Ura with Sdca

Similar to [24], the difference co-array (DCA) between two planar sub-arrays of a CPPA can be defined as

$$\begin{aligned} \mathbb{D}^1 &= \{p \mid p = p_1 - p_2, p_1 \in \mathbb{P}_1, p_2 \in \mathbb{P}_2\} \\ &= \{Nm_x + Nm_y - (Mn_x + Mn_y)\} \end{aligned} \tag{8}$$

$$\begin{aligned}\mathbb{D}^2 &= \{p \mid p = p_2 - p_1, p_1 \in \mathbb{P}_1, p_2 \in \mathbb{P}_2\} \\ &= \{Mn_x + Mn_y - (Nm_x + Nm_y)\}\end{aligned}\quad (9)$$

where $0 \leq m_x, m_y \leq 2M - 1, 0 \leq n_x, n_y \leq N - 1$. Then, we can define the sum-difference co-array (SDCA) as follows.

$$\mathbb{D} = \mathbb{D}^1 \cup \mathbb{D}^2 \quad (10)$$

From Equations (8) and (9), the maximum virtual aperture position of \mathbb{D}^1 and \mathbb{D}^2 in x -direction or y -direction are $(2MN - N)$ and $-(2MN - N)$. Similar to [24], the continuous lag ranges of \mathbb{D}^1 and \mathbb{D}^2 are $[-N + 1, MN + M - 1]$ and $[-(MN + M - 1), N - 1]$. Therefore, we have

$$\begin{aligned}2MN - N - (MN + M - 1) \\ &= MN - N - M + 1 \\ &= (M - 1)(N - 1) > 0\end{aligned}\quad (11)$$

where M and N are coprime integers greater than 1. From Equation (11), there are always entire rows and columns of missing elements for SDCA of CPPA with $M > 1$ and $N > 1$.

The cross-correlation matrix of the first and second planar sub-arrays can be expressed as

$$\begin{aligned}\hat{\mathbf{R}}_{12} &= 1/Q \sum_{q=1}^Q \mathbf{x}_1(q) \mathbf{x}_2^H(q) \\ &= \mathbf{A} \mathbf{R}_s \mathbf{B}^H + \mathbf{N}_{12}\end{aligned}\quad (12)$$

where Q is the total number of snapshots, $\mathbf{R}_s = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_K^2)$, σ_k^2 is the power of the k th signal and \mathbf{N}_{12} is the remaining residual term and can be expressed as

$$\begin{aligned}\mathbf{N}_{12} &= \frac{\mathbf{A}}{Q} \sum_{q=1}^Q \mathbf{s}(q) \mathbf{n}_2^H(q) + \\ &\sum_{q=1}^Q \mathbf{n}_1(q) \mathbf{s}^H(q) \frac{\mathbf{B}^H}{Q} + \frac{1}{Q} \sum_{q=1}^Q \mathbf{n}_1(q) \mathbf{n}_2^H(q)\end{aligned}\quad (13)$$

Then, we calculate the cross-correlation matrix of the second and the first planar sub-array, which can be given as

$$\begin{aligned}\hat{\mathbf{R}}_{21} &= 1/Q \sum_{q=1}^Q \mathbf{x}_2(q) \mathbf{x}_1^H(q) \\ &= \mathbf{B} \mathbf{R}_s \mathbf{A}^H + \mathbf{N}_{21}.\end{aligned}\quad (14)$$

where \mathbf{N}_{21} is the remaining residual term and can be written as

$$\begin{aligned}\mathbf{N}_{21} &= \frac{\mathbf{B}}{Q} \sum_{q=1}^Q \mathbf{s}(q) \mathbf{n}_1^H(q) + \\ &\sum_{q=1}^Q \mathbf{n}_2(q) \mathbf{s}^H(q) \frac{\mathbf{A}^H}{Q} + \frac{1}{Q} \sum_{q=1}^Q \mathbf{n}_2(q) \mathbf{n}_1^H(q).\end{aligned}\quad (15)$$

By extracting and rearranging $\hat{\mathbf{R}}_{12}$ and $\hat{\mathbf{R}}_{21}$, we can obtain a matrix $\mathbf{Z} \in \mathbb{C}^{n \times n}$ corresponding to a virtual URA, where $n = 2(2MN - N) + 1$ and its elements are located at the lag positions in the \mathbb{D} , as shown in Figure 2. Furthermore, we initialize the holes in \mathbf{Z} to zeros. From Figure 2, there are many missing elements in \mathbf{Z} . If we discard these discontinuous parts, it usually causes the loss of the virtual aperture of \mathbf{Z} . We introduce a matrix completion model to fill all holes to avoid virtual aperture loss.

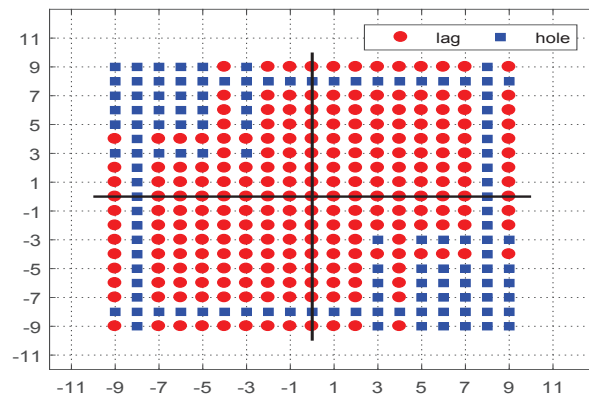


Figure 2. The structure of \mathbb{D} .

2.3. Co-Array Interpolation Initializes the Virtual Ura

Since there are missing elements in the entire rows and columns in \mathbf{Z} , the popular matrix completion methods cannot recover these holes [25].

The main reason is that the empty elements of the entire rows and columns will make the singular vector of \mathbf{Z} more sparse, thereby reducing the recovery performance. To solve this problem, we first extract a DCA from the y direction in \mathbf{Z} , thereby obtaining a virtual ULA \mathbf{y}_1 from $(0, -(2MN - N)d)$ to $(0, 2MN - N)d$, where the holes at positions $(0, -(2MN - N - 1)d)$ and $(0, 2MN - N - 1)d$ are initialized to zeros. Similarly, we can also extract a DCA from the x direction in \mathbf{Z} , getting a virtual ULA \mathbf{y}_2 from $(-(2MN - N), 0)d$ to $(2MN - N, 0)d$, where the holes at positions $(-(2MN - N - 1), 0)d$ and $(2MN - N - 1, 0)d$ are initialized to zeros.

Thus, \mathbf{y}_1 and \mathbf{y}_2 can be represented as

$$\mathbf{y}_1 = \bar{\mathbf{A}}\mathbf{s} + \mathbf{e}_1 \tag{16}$$

$$\mathbf{y}_2 = \bar{\mathbf{B}}\mathbf{s} + \mathbf{e}_2 \tag{17}$$

where \mathbf{e}_1 and \mathbf{e}_2 are error terms, $\bar{\mathbf{A}} = [\bar{\mathbf{a}}(u_1), \bar{\mathbf{a}}(u_2), \dots, \bar{\mathbf{a}}(u_K)]$, $\bar{\mathbf{B}} = [\bar{\mathbf{b}}(v_1), \bar{\mathbf{b}}(v_2), \dots, \bar{\mathbf{b}}(v_K)]$. $\bar{\mathbf{a}}(u_k)$ and $\bar{\mathbf{b}}(v_k)$ can be represented as

$$\bar{\mathbf{a}}(u_k) = \begin{bmatrix} e^{-j\pi(L-1)u_k}, 0, e^{-j\pi(L-3)u_k}, \dots, e^{-j\pi u_k} \\ 1, e^{j\pi u_k}, \dots, e^{j\pi(L-3)u_k}, 0, e^{j\pi(L-1)u_k} \end{bmatrix}^T \tag{18}$$

$$\bar{\mathbf{b}}(v_k) = \begin{bmatrix} e^{-j\pi(L-1)v_k}, 0, e^{-j\pi(L-3)v_k}, \dots, e^{-j\pi v_k} \\ 1, e^{j\pi v_k}, \dots, e^{j\pi(L-3)v_k}, 0, e^{j\pi(L-1)v_k} \end{bmatrix}^T \tag{19}$$

where $L = (n + 1)/2$.

To fill the holes of the virtual ULA, we usually divide \mathbf{y}_1 and \mathbf{y}_2 into L sub-arrays to compute a smooth covariance matrix with Hermitian positive semidefinite (PSD) Toeplitz structure. According to [17], we can build two covariance matrices of virtual signals \mathbf{y}_1 and \mathbf{y}_2 and they can be expressed as

$$\tilde{\mathbf{R}}_1 = \begin{bmatrix} \langle \mathbf{y}_1 \rangle_L & \langle \mathbf{y}_1 \rangle_{L+1}^* & \cdots & \langle \mathbf{y}_1 \rangle_{2L-1}^* \\ \langle \mathbf{y}_1 \rangle_{L+1} & \langle \mathbf{y}_1 \rangle_L & \cdots & \langle \mathbf{y}_1 \rangle_{2L-2}^* \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{y}_1 \rangle_{2L-1} & \langle \mathbf{y}_1 \rangle_{2L-2} & \cdots & \langle \mathbf{y}_1 \rangle_L \end{bmatrix} \tag{20}$$

$$\tilde{\mathbf{R}}_2 = \begin{bmatrix} \langle \mathbf{y}_2 \rangle_L & \langle \mathbf{y}_2 \rangle_{L+1}^* & \cdots & \langle \mathbf{y}_2 \rangle_{2L-1}^* \\ \langle \mathbf{y}_2 \rangle_{L+1} & \langle \mathbf{y}_2 \rangle_L & \cdots & \langle \mathbf{y}_2 \rangle_{2L-2}^* \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{y}_2 \rangle_{2L-1} & \langle \mathbf{y}_2 \rangle_{2L-2} & \cdots & \langle \mathbf{y}_2 \rangle_L \end{bmatrix} \quad (21)$$

$\tilde{\mathbf{R}}_1$ and $\tilde{\mathbf{R}}_2$ have holes corresponding to the zeros of the virtual array signals \mathbf{y}_1 and \mathbf{y}_2 . Therefore, we build the binary matrix $\mathbf{D} \in \mathbb{R}^{L \times L}$ to distinguish virtual array signal positions and holes, where the elements of \mathbf{D} corresponding to non-zero signal positions are ones, and the rest are zeros.

To fill the empty elements of $\tilde{\mathbf{R}}_1$ and $\tilde{\mathbf{R}}_2$, we formulate the nuclear norm minimization optimization problem as

$$\min \|\mathbf{R}_1\|_{**}, \text{ s.t. } \|(\tilde{\mathbf{R}}_1 - \mathbf{R}_1) \circ \mathbf{D}\|_F \leq \delta_1 \quad (22)$$

$$\min \|\mathbf{R}_2\|_{**}, \text{ s.t. } \|(\tilde{\mathbf{R}}_2 - \mathbf{R}_2) \circ \mathbf{D}\|_F \leq \delta_2 \quad (23)$$

where δ_1 and δ_2 are the thresholds of the fitting error, and $\|\cdot\|_*$ represents the nuclear norm of a matrix, which is the relaxation of the rank norm of the matrix.

Since the optimization problems (22) and (23) are convex, we can solve them directly by employing the interior point method [28]. Then, we can obtain the estimated optimal covariance matrices $\hat{\mathbf{R}}_1$ and $\hat{\mathbf{R}}_2$ corresponding to $\tilde{\mathbf{R}}_1$ and $\tilde{\mathbf{R}}_2$. Subsequently, we extract the elements $\hat{\mathbf{R}}_1(1, L-2)$ and $\hat{\mathbf{R}}_1(L-2, 1)$, filling the holes at positions $(0, -(L-2)d)$ and $(0, L-2)d$ in matrix \mathbf{Z} . Similarly, we use $\hat{\mathbf{R}}_2(L-2, 1)$ and $\hat{\mathbf{R}}_2(1, L-2)$ to fill the holes of matrix \mathbf{Z} at positions $(-(L-2), 0)d$ and $(L-2, 0)d$. After the above interpolation operation, we can initialize the matrix \mathbf{Z} to obtain a matrix $\bar{\mathbf{Z}}$ that does not contain the missing elements of the entire rows and columns. In addition, we demonstrate the structure of the virtual signal position set \mathbb{D} corresponding to $\bar{\mathbf{Z}}$, as shown in Figure 3.

From Figure 3, $\bar{\mathbf{Z}}$ can be represented as a single-snapshot signal model received by a virtual URA and can be written as

$$\bar{\mathbf{Z}} = (\bar{\mathbf{A}}\mathbf{R}_s\bar{\mathbf{B}}^T + \mathbf{E}) \circ \mathbf{S} \quad (24)$$

where $\mathbf{E} \in \mathbb{C}^{n \times n}$ is the error term, and $\mathbf{S} \in \mathbb{R}^{n \times n}$ is a binary matrix whose elements correspond to holes in $\bar{\mathbf{Z}}$ as zeros, and the rest are ones.

2.4. Admm-Based Enhanced Matrix Completion

After the above co-array interpolation processing, there are no missing elements in the entire rows and columns in $\bar{\mathbf{Z}}$, which prompts us to use matrix completion to fill the remaining empty elements. Furthermore, we assume that $\bar{\mathbf{Z}}$ has a low-rank property because the number of targets is usually smaller than the dimension of $\bar{\mathbf{Z}}$. Based on these conditions, the matrix completion model can be expressed as

$$\min_{\mathbf{Y}} \|\mathbf{Y}\|_{**}, \text{ s.t. } \|U_{\Omega}(\mathbf{Y} - \bar{\mathbf{Z}})\|_F \leq \alpha \quad (25)$$

where α is the threshold of the fitting error, Ω is a set of positions of the known entries of $\bar{\mathbf{Z}}$, and the matrix operator U_{Ω} can be defined as

$$(U_{\Omega}(\mathbf{Y}))_{ij} = \begin{cases} \mathbf{Y}_{ij}, & \text{if } (i, j) \in \Omega \\ 0, & \text{otherwise} \end{cases} \quad (26)$$

The optimization problem (25) does not use the TNNR-based matrix completion method proposed in [24]. The reason is that the TNNR method needs to know the number of targets, but, in practical problems, the number of targets is usually unknown. In addition, the optimization problem (25) is convex and can be solved by the interior point method [28]. However, the optimization model (25) does not take the inherent shift-invariant structure of $\bar{\mathbf{Z}}$, reducing the accuracy of matrix completion. What is worse, when the number of

targets is larger than the dimension of $\bar{\mathbf{Z}}$, this model always fails to recover the completion matrix \mathbf{Y} .

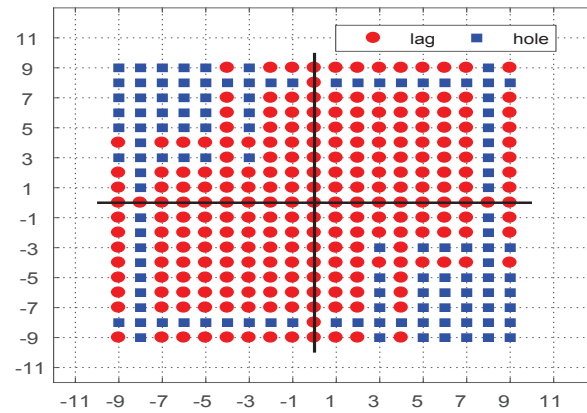


Figure 3. The structure of $\bar{\mathbb{D}}$.

In order to solve these problems, we propose an ADMM-based, enhanced matrix-completion algorithm. The proposed method makes full use of the shift-invariant structure of $\bar{\mathbf{Z}}$ to form the enhanced matrix \mathbf{M}_e with a two-fold Hankel structure, thereby improving the performance of matrix completion. In addition, we also developed a fast ADMM algorithm to reduce the computational complexity of the NNM-based algorithm and named the algorithm EMaC-ADMM. According to [29], the enhanced matrix \mathbf{M}_e corresponding to the matrix $\bar{\mathbf{Z}}$ can be defined as

$$\mathbf{M}_e = \begin{bmatrix} \mathbf{M}_0 & \mathbf{M}_1 & \cdots & \mathbf{M}_{n_1-k_1} \\ \mathbf{M}_1 & \mathbf{M}_2 & \cdots & \mathbf{M}_{n_1-k_1+1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{M}_{k_1-1} & \mathbf{M}_{k_1} & \cdots & \mathbf{M}_{n_1-1} \end{bmatrix} \tag{27}$$

where $k_1 = \frac{n+1}{2}$ is a pencil parameter. The i th block Hankel matrix $\mathbf{M}_i \in \mathbb{C}^{k_2 \times (n-k_2+1)}$ contained in \mathbf{M}_e can be expressed as

$$\mathbf{M}_i = \begin{bmatrix} \bar{\mathbf{Z}}_{(i,0)} & \bar{\mathbf{Z}}_{(i,1)} & \cdots & \bar{\mathbf{Z}}_{(i,n-k_2)} \\ \bar{\mathbf{Z}}_{(i,1)} & \bar{\mathbf{Z}}_{(i,2)} & \cdots & \bar{\mathbf{Z}}_{(i,n-k_2+1)} \\ \vdots & \vdots & \vdots & \vdots \\ \bar{\mathbf{Z}}_{(i,k_2-1)} & \bar{\mathbf{Z}}_{(i,k_2)} & \cdots & \bar{\mathbf{Z}}_{(i,n-1)} \end{bmatrix} \tag{28}$$

where $k_2 = \frac{n+1}{2}$ is another pencil parameter.

Then, we can recover the missing elements of the noise-free enhanced matrix \mathbf{M}_e by forming the matrix completion model as follows.

$$\begin{aligned} \min & \|\mathbf{Y}_e\|_* \\ \text{s.t.} & \|U_{\Omega_e}(\mathbf{Y}_e)\|_F = \|U_{\Omega_e}(\mathbf{M}_e)\|_F \end{aligned} \tag{29}$$

where Ω_e refers to the set of known element positions of the enhanced matrix \mathbf{M}_e , $\mathbf{Y}_e \in \mathbb{C}^{m \times m}$ is the enhanced form of $\mathbf{Y} \in \mathbb{C}^{n \times n}$, and $m = k_1 k_2$. However, in practical applications, the existence of noise is inevitable. Therefore, the enhanced matrix completion model (29) for bounded white Gaussian noise can be expressed as

$$\min_{\mathbf{Y}_e} \|\mathbf{Y}_e\|_*, \text{ s.t. } \|U_{\Omega_e}(\mathbf{Y}_e - \mathbf{M}_e)\|_F \leq \delta \tag{30}$$

where $\delta > 0$ is a parameter to constrain the noise. The optimization problem (30) can be efficiently solved by the interior point method [28]. However, the interior point method

takes up large memory when completing large-scale matrices, reducing the running speed of the algorithm.

Given the superior convergence performance and fast convergence speed of the ADMM framework, we propose an EMaC-ADMM method to solve the optimization problem (30). Therefore, we rewrite the optimization problem (30) as

$$\begin{aligned} & \min_{\mathbf{Y}_e, \mathbf{N}_e, \mathbf{W}} \|\mathbf{Y}_e\|_* \\ & \text{s.t. } \mathbf{Y}_e + \mathbf{N}_e = \mathbf{W} \\ & U_{\Omega_e}(\mathbf{W}) = U_{\Omega_e}(\mathbf{M}_e) \\ & \|U_{\Omega_e}(\mathbf{N}_e)\|_F \leq \sigma \end{aligned} \tag{31}$$

where $\mathbf{N}_e \in \mathbb{C}^{m \times m}$ refers to the enhanced form of the noise matrix $\mathbf{N} \in \mathbb{C}^{n \times n}$, and $\mathbf{W} \in \mathbb{C}^{m \times m}$ is an additional variable.

The augmented Lagrangian function of (31) is defined as

$$L_\beta(\mathbf{Y}_e, \mathbf{N}_e, \Lambda) = \|\mathbf{Y}_e\|_* + \langle \Lambda, \mathbf{Y}_e + \mathbf{N}_e - \mathbf{W} \rangle + \frac{\beta}{2} \|\mathbf{Y}_e + \mathbf{N}_e - \mathbf{W}\|_F^2 \tag{32}$$

where $\Lambda \in \mathbb{C}^{m \times m}$ are the Lagrange multipliers (dual variable), and $\beta > 0$ denotes the penalty parameter. We utilize the ADMM framework to iteratively update the primal optimal variables $\mathbf{Y}_e, \mathbf{N}_e, \mathbf{W}$, and the dual variable Λ by finding the saddle point of the augmented Lagrange. In the k th iteration, the specific execution process of ADMM can be expressed as

$$\mathbf{Y}_e^{k+1} = \arg \min_{\mathbf{Y}_e} L_\beta(\mathbf{Y}_e, \mathbf{N}_e^k, \mathbf{W}^k, \Lambda^k) \tag{33}$$

$$\mathbf{N}_e^{k+1} = \arg \min_{\mathbf{N}_e} L_\beta(\mathbf{Y}_e^{k+1}, \mathbf{N}_e, \mathbf{W}^k, \Lambda^k) \tag{34}$$

$$\mathbf{W}^{k+1} = \arg \min_{U_{\Omega_e}(\mathbf{W})=U_{\Omega_e}(\mathbf{M}_e)} L_\beta(\mathbf{Y}_e^{k+1}, \mathbf{N}_e^{k+1}, \mathbf{W}, \Lambda^k) \tag{35}$$

$$\Lambda^{k+1} = \Lambda^k + \beta U_{\Omega_e}(\mathbf{Y}_e^{k+1} + \mathbf{N}_e^{k+1} - \mathbf{W}^{k+1}) \tag{36}$$

For problems (33) to (36), ADMM updates $\mathbf{Y}_e^{k+1}, \mathbf{N}_e^{k+1}, \mathbf{W}^{k+1}$ and Λ^{k+1} in an alternate iterative manner, where the dual variable Λ^{k+1} is updated by the gradient ascent method. For sub-problem (33), by ignoring the constant term, \mathbf{Y}_e^{k+1} can be updated as

$$\mathbf{Y}_e^{k+1} = \arg \min_{\mathbf{Y}_e} \|\mathbf{Y}_e\|_* + \frac{\beta}{2} \left\| \mathbf{Y}_e + \mathbf{N}_e^k - \mathbf{W}^k + \frac{\Lambda^k}{\beta} \right\|_F^2 \tag{37}$$

The convex optimization problem (37) can be efficiently solved by the singular value shrinkage operator [30], and we have

$$\mathbf{Y}_e^{k+1} = \mathbf{U} \text{diag} \left(\max \left\{ \sigma_i - \frac{1}{\beta}, 0 \right\} \right) \mathbf{V}^H \tag{38}$$

where σ_i, \mathbf{U} and \mathbf{V} are the i th singular value, left and right singular vectors of $\mathbf{M}_e - \mathbf{N}_e^k - \frac{\Lambda^k}{\beta}$, respectively.

Given \mathbf{Y}_e^{k+1} and ignoring the irrelevant terms of \mathbf{N}_e , sub-problem (34) can be rewritten as

$$\begin{aligned} & \mathbf{N}_e^{k+1} = \arg \min_{\mathbf{N}_e} \|\mathbf{N}_e - \mathbf{C}\|_F^2 \\ & \text{s.t. } \|U_{\Omega_e}(\mathbf{N}_e)\|_F \leq \delta \end{aligned} \tag{39}$$

where $\mathbf{C} = (\mathbf{W}^k - \mathbf{Y}_e^{k+1} - \Lambda^k / \beta)$. The solution to the optimization problem (39) can be divided into two cases. In the first case, when $\|U_{\Omega_e}(\mathbf{C})\|_F \leq \delta$, the solution of (39) can be given directly

$$\mathbf{N}_e^{k+1} = U_{\Omega_e}(\mathbf{C}) \quad (40)$$

For the second case, if $\|U_{\Omega_e}(\mathbf{C})\|_F > \delta$, the constraints of the optimization problem (39) can be rewritten as equality constraints [31] and can be expressed as

$$\begin{aligned} \mathbf{N}_e^{k+1} &= \arg \min_{\mathbf{N}_e} \|\mathbf{N}_e - \mathbf{C}\|_F^2 \\ \text{s.t. } &\|U_{\Omega_e}(\mathbf{N}_e)\|_F = \delta \end{aligned} \quad (41)$$

therefore, the solution of optimization problem (41) can be given

$$\mathbf{N}_e^{k+1} = \frac{\delta U_{\Omega_e}(\mathbf{C})}{\|U_{\Omega_e}(\mathbf{C})\|_F} \quad (42)$$

Fixing \mathbf{Y}_e^{k+1} and \mathbf{N}_e^{k+1} and ignoring the irrelevant terms of \mathbf{W} , (35) can be rewritten as

$$\mathbf{W}^{k+1} = \arg \min_{U_{\Omega_e}(\mathbf{W})=U_{\Omega_e}(\mathbf{M}_e)} \frac{\beta}{2} \left\| \mathbf{W} - (\mathbf{Y}_e^{k+1} + \mathbf{N}_e^{k+1} + \frac{\Lambda^k}{\beta}) \right\|_F^2 \quad (43)$$

for (43), \mathbf{W}^{k+1} has a closed-form solution and can be expressed as

$$\mathbf{W}^{k+1} = \mathbf{Y}_e^{k+1} + \mathbf{N}_e^{k+1} + \frac{\Lambda^k}{\beta} \quad (44)$$

Then, we add constraints on \mathbf{W} and fix the known elements of \mathbf{M}_e , and finally update \mathbf{W}^{k+1} as

$$\mathbf{W}^{k+1} = U_{\Omega_e^c}(\mathbf{W}^{k+1}) + U_{\Omega_e}(\mathbf{M}_e) \quad (45)$$

where Ω_e^c refers to the set of positions of the unobserved elements of the enhanced matrix \mathbf{M}_e .

The specific execution process of EMaC-ADMM is summarized in Algorithm 1. EMaC-ADMM utilizes the potential shift-invariance structure of $\bar{\mathbf{Z}}$ to improve the matrix completion performance. In addition, it also employs the ADMM framework to reduce the computational complexity of large-scale matrix completion problems caused by matrix enhancement.

Algorithm 1 EMaC-ADMM

- 1: **Input:** $\mathbf{M}_e, \Omega_e, \varepsilon_1, \delta, \beta > 0$;
 - 2: **Initialization:** $k = 1, \mathbf{Y}_e^1 = \mathbf{M}_e, \mathbf{N}_e^1 = \mathbf{0}, \mathbf{W}_e^1 = \mathbf{M}_e, \Lambda^{k+1} = \mathbf{M}_e$.
 - 3: **repeat**
 - 4: Update \mathbf{Y}_e^{k+1} via (37).
 - 5: Update \mathbf{N}_e^{k+1} via (39).
 - 6: Update \mathbf{W}^{k+1} via (43).
 - 7: Update the Lagrange multiplier Λ^{k+1} via (36).
 - 8: $k = k + 1$.
 - 9: **until** $\|\mathbf{Y}_e^{k+1} - \mathbf{Y}_e^k\|_F < \varepsilon_1$.
 - 10: **Output** $\hat{\mathbf{Y}}_e = \mathbf{Y}_e^k$.
-

2.5. Sparse Matrix-Recovery-Based 2-D Doa Estimation

After the enhanced matrix completion in the previous sub-section, we can recover the completion matrix $\hat{\mathbf{Y}}$ from the enhanced form $\hat{\mathbf{Y}}_e$, and it can be represented as

$$\hat{\mathbf{Y}} = \tilde{\mathbf{A}}\mathbf{R}_s\tilde{\mathbf{B}}^T + \mathbf{E} \quad (46)$$

where \mathbf{E} is the error term.

To estimate the 2-D DOA of targets, we establish the following l_1 -norm-based sparse matrix recovery model.

$$\min_{\mathbf{X}} \left\{ \frac{1}{2} \|\hat{\mathbf{Y}} - \tilde{\mathbf{A}}\mathbf{X}\tilde{\mathbf{B}}^T\|_F^2 + \rho \|\mathbf{X}\|_1 \right\} \quad (47)$$

where ρ is the regularization parameter, $\mathbf{X} \in \mathbb{C}^{I \times J}$ is an unknown sparse matrix, $\tilde{\mathbf{A}} = [\tilde{\mathbf{a}}(u_1), \tilde{\mathbf{a}}(u_2), \dots, \tilde{\mathbf{a}}(u_I)]$ and $\tilde{\mathbf{B}} = [\tilde{\mathbf{b}}(v_1), \tilde{\mathbf{b}}(v_2), \dots, \tilde{\mathbf{b}}(v_J)]$ are overcomplete dictionary matrices in the transform domain $u \in [0, 1]$ and $v \in [-1, 1]$, respectively, and I and J represent the number of sampling grids.

The optimization problem (47) can be transformed into a sparse recovery problem in vector form and solved by the interior point method [28]. However, the Kronecker product operation between dictionary matrices will significantly increase the dimension of the dictionary matrix, which increases the computational burden of the vector-form, sparse recovery algorithm.

To reduce the computational burden of the traditional vector-form, sparse recovery algorithm, we utilize the FISTA method [32] to perform sparse matrix recovery. However, the FISTA method only works on real-valued data and cannot be directly applied to complex-valued data. Therefore, we derive a complex-valued FISTA algorithm named (CV-FISTA). The specific derivation process is as follows.

Similar to [32], we rewrite the optimization problem (47) as

$$\min_{\mathbf{X}} H(\mathbf{X}) + \rho F(\mathbf{X}) \quad (48)$$

where $H(\mathbf{X}) = \frac{1}{2} \|\hat{\mathbf{Y}} - \tilde{\mathbf{A}}\mathbf{X}\tilde{\mathbf{B}}^T\|_F^2$ is a smooth convex function, $F(\mathbf{X}) = \|\mathbf{X}\|_1$ is a non-smooth convex function. To solve the optimization problem (48), we apply a quadratic approximation of $H(\mathbf{X})$ at a given point \mathbf{P} .

$$Q_L(\mathbf{X}, \mathbf{P}) = H(\mathbf{P}) + \text{tr}(\nabla H(\mathbf{P})^T(\mathbf{X} - \mathbf{P})) + \frac{L_f}{2} \|\mathbf{X} - \mathbf{P}\|_F^2 + \rho F(\mathbf{X}) \quad (49)$$

where $L_f = \|\tilde{\mathbf{B}} \otimes \tilde{\mathbf{A}}\|_F^2$ is the Lipschitz constant. After some simplification, we can rewrite (49) as

$$Q_L(\mathbf{X}, \mathbf{P}) = \frac{L_f}{2} \|\mathbf{X} - U(\mathbf{P})\|_F^2 + \rho \|\mathbf{X}\|_1 + C(\mathbf{P}) \quad (50)$$

where $C(\mathbf{P})$ is a constant term concerning \mathbf{P} , and $U(\mathbf{P})$ can be expressed as

$$U(\mathbf{P}) = \mathbf{P} - \frac{1}{L_f} \tilde{\mathbf{A}}^H (\tilde{\mathbf{A}}\mathbf{X}\tilde{\mathbf{B}}^T - \hat{\mathbf{Y}}) \tilde{\mathbf{B}}^* \quad (51)$$

Proof: please see Appendix A.

Ignoring the constant term, we rewrite the optimization problem (50) as

$$\arg \min_{\mathbf{X}} Q_L(\mathbf{X}, \mathbf{P}) = \arg \min_{\mathbf{X}} \left\{ \frac{L_f}{2} \|\mathbf{X} - U(\mathbf{P})\|_F^2 + \rho \|\mathbf{X}\|_1 \right\} \quad (52)$$

The optimization problem (52) can be solved by the soft threshold method [33] and can be expressed as

$$\arg \min_{\mathbf{X}} Q_L(\mathbf{X}, \mathbf{P}) = \text{soft} \left(U(\mathbf{P}), \frac{\rho}{L_f} \right) \quad (53)$$

where $\text{soft}(\cdot)$ can be defined as

$$\text{soft}(\mathbf{X}, c) = \mathbf{X}_{ij} \left(1 - \frac{c}{|\mathbf{X}_{ij}|} \right)_+ \quad (54)$$

where $c > 0$ is a constant and \mathbf{X}_{ij} is the (i, j) th element of matrix \mathbf{X} .

We give the specific execution process of the CV-FISTA algorithm in Algorithm 2. After several iterations, we can get the sparse recovery matrix $\hat{\mathbf{X}}$, and the azimuth and elevation angles can be obtained by calculating

$$\hat{\varphi}_k = \tan^{-1}(\hat{v}_k/\hat{u}_k), \quad \hat{\theta}_k = \sin^{-1}\sqrt{\hat{u}_k^2 + \hat{v}_k^2} \quad (55)$$

where \hat{u}_k and \hat{v}_k are the estimated parameters to obtain the k th target from $\hat{\mathbf{X}}$.

Algorithm 2 CV-FISTA

- 1: **Input:** the completion matrix $\hat{\mathbf{Y}}$; overcomplete dictionary matrices $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$; tolerance ε_2 ; the regularization parameter $\rho > 0$.
 - 2: **Initialization:** $k = 1, \mathbf{X}_1 = \mathbf{0}, \mathbf{X}_0 = \mathbf{0}, t_1 = 1, t_0 = 1$.
 - 3: **repeat**
 - 4: $\mathbf{P}_k = \mathbf{X}_k + \frac{t_{k-1}-1}{t_k}(\mathbf{X}_k - \mathbf{X}_{k-1})$.
 - 5: $U(\mathbf{P}_k) = \mathbf{P}_k - \frac{1}{L_f} \tilde{\mathbf{A}}^H (\tilde{\mathbf{A}} \mathbf{P}_k \tilde{\mathbf{B}}^T - \hat{\mathbf{Y}}) \tilde{\mathbf{B}}^*$.
 - 6: $\mathbf{X}_{k+1} = \text{soft}\left(U(\mathbf{P}_k), \frac{\rho}{L_f}\right)$.
 - 7: $t_{k+1} = (1 + \sqrt{4t_k^2 + 1})/2$.
 - 8: $k = k + 1$.
 - 9: **until** $\|\mathbf{X}_{k+1} - \mathbf{X}_k\|_F < \varepsilon_2$.
 - 10: **Output** $\hat{\mathbf{X}} = \mathbf{X}_k$.
-

2.6. Complexity Analysis

We analyze the computational complexity of the proposed algorithm and compare it with state-of-the-art algorithms. The computational complexity of the proposed algorithm mainly includes two parts: the first part is matrix completion, and the second part is sparse matrix recovery. For the first part, the computational complexity of co-array interpolation initialization and enhanced matrix completion is about $\mathcal{O}(L^3K_1 + m^3K_2)$, where m is the dimension of the enhanced matrix \mathbf{Y}_e , K_1 is the number of iterations of co-array interpolation, and K_2 is the number of iterations of the EMaC-ADMM algorithm. Then, we analyze the computational complexity of sparse matrix recovery, mainly caused by matrix multiplication, which is approximate $\mathcal{O}(I^3K_3)$, where I is the number of the searching grid and K_3 is the number of iterations of the CV-FISTA algorithm.

Therefore, the total computational complexity of the proposed algorithm is about $\mathcal{O}(L^3K_1 + m^3K_2 + I^3K_3)$. Subsequently, we analyze the computational complexity of the TNNR algorithm [20], which is mainly caused by singular value decomposition, about $\mathcal{O}(rn^2K_4)$, where K_4 is the number of iterations, and r is the rank of the matrix. Then, the CV-FISTA method is used to estimate the angle of targets. Therefore, the final computational complexity of the TNNR algorithm is about $\mathcal{O}(rn^2K_4 + I^3K_3)$.

Next, we analyze the computational complexity of the partial spectral search (PSS) algorithm [23] and the iterative sparse recovery (ISR) algorithm [24]. The computational complexity of the PSS algorithm is mainly caused by spectral search and eigenvalue

decomposition, which is about $\mathcal{O}\left(I\left(\frac{(2M)^4}{N^2} + \frac{N^4}{(2M)^2}\right) + (2M)^3 + N^3\right)$. Since I is much larger than M and N , the total computational complexity of the PSS algorithm is about $\mathcal{O}\left(I\left(\frac{(2M)^4}{N^2} + \frac{N^4}{(2M)^2}\right)\right)$. The computational complexity of the ISR algorithm is approximately $\mathcal{O}(r^3 I^3 K_5 + I^3)$, where K_5 is the number of iterations. We clearly describe the computational complexity of the above algorithms in Table 2.

Table 2. Computation complexity comparison of different algorithms.

Proposed algorithm	$\mathcal{O}(L^3 K_1 + m^3 K_2 + I^3 K_3)$
TNNR in [20]	$\mathcal{O}(rn^2 K_4 + I^3 K_3)$
PSS in [23]	$\mathcal{O}\left(I\left(\frac{(2M)^4}{N^2} + \frac{N^4}{(2M)^2}\right)\right)$
ISR in [24]	$\mathcal{O}(r^3 I^3 K_5 + I^3)$

Finally, we compare the computational complexity of the above algorithms versus the number of sensors in Figure 4, where M is fixed at 2, $N = [3, 5, 7, 9]$. As shown in Figure 4, the computational complexity of the proposed algorithm is higher than the TNNR algorithm. The cause is that, in the process of matrix completion, the proposed algorithm needs the co-array interpolation initialization process, so it has higher computational complexity than the TNNR algorithm. The PSS algorithm has the lowest computational complexity because it does not require iterative operations. When $I \gg M, N, L$, and m , the computational complexity of the proposed algorithm is less than the ISR algorithm. However, as the total number of sensors increases, the proposed algorithm's computational complexity increases faster than the other tested algorithms. Therefore, when the number of sensors is large, the computational complexity of the proposed algorithm is higher than that of the ISR algorithm. The reason is that the proposed algorithm's enhanced matrix further expands the dimension of the virtual URA, causing the complexity to increase.

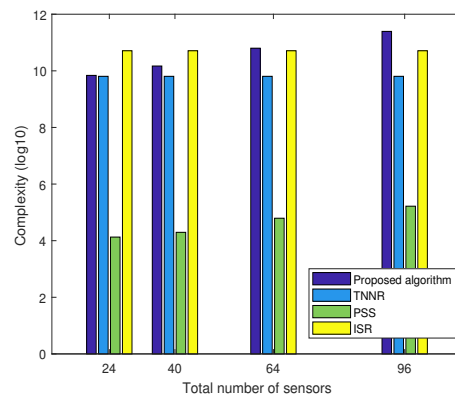


Figure 4. Complexity comparison versus the total number of sensors.

3. Results

In our simulations, we consider a CPPA with $M = 2$ and $N = 3$. The parameters $\delta = 1$, $\delta_1 = 1$, $\delta_2 = 1$, $\beta = 0.5$, and $\rho = 1e^2$. After matrix completion processing, we can finally obtain a virtual URA with virtual sensor positions from $(-9, -9)d$ to $(9, 9)d$. We compare the 2-D DOA estimation performance of the proposed algorithm with several recently proposed algorithms, including the TNNR algorithm [20], the ISR algorithm [24] and the PSS algorithm [23]. In addition, the Cramer–Rao bound (CRB) [24] is used as a reference. Finally, the root mean square error (RMSE) is utilized to compare the angle estimation performance of the above test algorithm and can be defined as

$$\text{RMSE} = \sqrt{\frac{1}{TK} \sum_{t=1}^T \sum_{k=1}^K [(\varphi_k - \hat{\varphi}_{k,t})^2 + (\theta_k - \hat{\theta}_{k,t})^2]} \quad (56)$$

where $(\hat{\varphi}_{k,t}, \hat{\theta}_{k,t})$ denotes the estimate of k th target (φ_k, θ_k) in the t th Monte Carlo trial, $T = 100$ is the number of Monte Carlo trials.

In the first experiment, we compare the number of degrees of freedom (DOF) of the proposed algorithm with the other tested algorithms. After the enhanced matrix completion, the proposed algorithm has 361 DOFs corresponding to the matrix $\hat{\mathbf{Y}}$. Since the ISR algorithm utilizes the consecutive part of the SDCA, it has 175 DOFs. A standard URA consisting of 24 physical sensors has 24 DOFs. In addition, the PSS algorithm has only 9 DOFs. Therefore, the proposed algorithm has the maximum DOFs compared to other algorithms. Figure 5a depicts the contour map of the normalized spectrum of the proposed algorithm, where the signal-to-noise ratio (SNR) is 0 dB, the number of snapshots is 500, and the searching grid is 0.002 for $u \in [0, 1]$ and $v \in [-1, 1]$. As shown in Figure 5a, the proposed algorithm can resolve all 27 targets using only 24 physical sensors. Figure 5b shows the contour map of the normalized spectrum of the TNNR algorithm. From Figure 5b, the TNNR algorithm can resolve up to 16 targets. This is because the rank is limited by the dimension of the matrix \mathbf{Z} , thereby reducing the maximum number of resolved targets of the TNNR algorithm. Figure 5c depicts the standard URA 2-D DOA estimation, where the 2-D unitary ESPRIT algorithm [7] is utilized. From Figure 5d, the standard URA can resolve up to eight targets under low SNR conditions. Figure 5d shows the 2-D DOA estimation of the ISR algorithm. Since the ISR algorithm only utilizes 15 DOFs for the initialization process of u or v , the maximum number of targets detected is limited. Therefore, the virtual aperture expansion ability of the proposed algorithm is better than that of the TNNR algorithm, ISR algorithm, PSS algorithm, and standard URA.

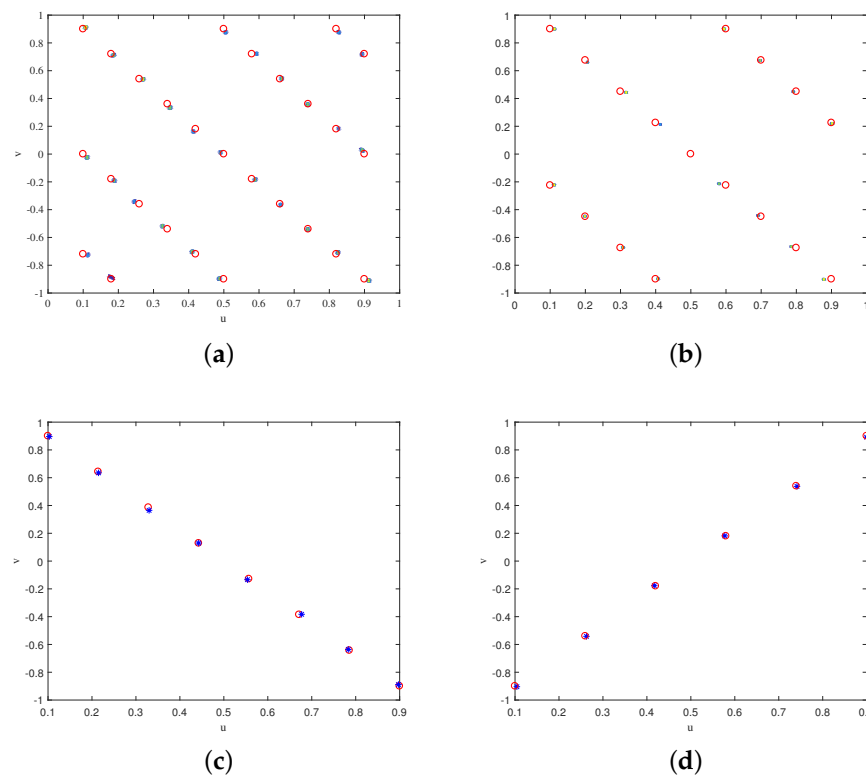


Figure 5. (a) Contour map of the proposed algorithm. (b) Contour map of the TNNR algorithm. (c) Estimated 2-D DOA with standard URA. (d) Estimated 2-D DOA of the ISR algorithm.

In the second experiment, we evaluate the estimation performance of all tested algorithms for uncorrelated targets. Figure 6 shows the RMSE of the angle estimate versus SNR, where two uncorrelated targets are located at $(\theta_1, \varphi_1) = (10^\circ, 10^\circ)$ and $(\theta_2, \varphi_2) = (30^\circ, 30^\circ)$, the number of snapshots is 10, and the SNR varies from 0 dB to 10 dB. As shown in Figure 6, when the SNR is greater than or equal to 4 dB, the proposed algorithm has the best estimation performance compared to the other test algorithms. The reason is that the proposed algorithm utilizes virtual aperture extension and sparse matrix recovery. However, when the SNR is lower than 4 dB, the performance of the ISR algorithm is better than that of the proposed algorithm because the matrix completion accuracy of the proposed algorithm decreases, leading to decrease in the estimation accuracy of the proposed algorithm. The performance of the TNNR algorithm degrades when the SNR is high. The cause is that the TNNR algorithm cannot complete the missing elements of the entire rows and columns in \mathbf{Z} , which significantly affects the sparse matrix recovery accuracy and reduces the DOA estimation performance. The PSS algorithm has the worst estimation performance due to not utilizing virtual aperture expansion. Furthermore, as the SNR increases, the estimated performance of all tested algorithms becomes relatively flat, which is limited by the pre-defined sampling grid.

In the third experiment, we compare the estimated performance of all tested algorithms versus the number of snapshots, where we assume two uncorrelated targets located at $(\theta_1, \varphi_1) = (10^\circ, 10^\circ)$ and $(\theta_2, \varphi_2) = (30^\circ, 30^\circ)$, the SNR is 0 dB, and the number of snapshots varies from 10 to 50. As demonstrated in Figure 7, when the number of snapshots exceeds 20, the proposed algorithm can obtain the optimal estimation performance due to the virtual aperture expansion and sparse matrix recovery. When the number of snapshots is less than or equal to 20, the ISR algorithm is better than the other test algorithms because it uses an iterative sparse recovery algorithm to improve the angle estimation performance. Under the condition of a low SNR and low snapshots, the TNNR algorithm outperforms the proposed algorithm. The explanation is that the TNNR algorithm has better completion accuracy than the proposed algorithm for empty elements that are not entire rows and columns. Therefore, the proposed algorithm has better estimation performance than the other test algorithms under mild conditions.

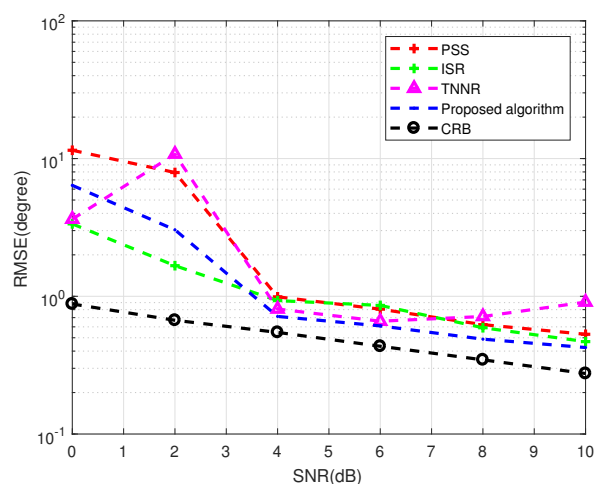


Figure 6. RMSE versus the SNR for two uncorrelated targets.

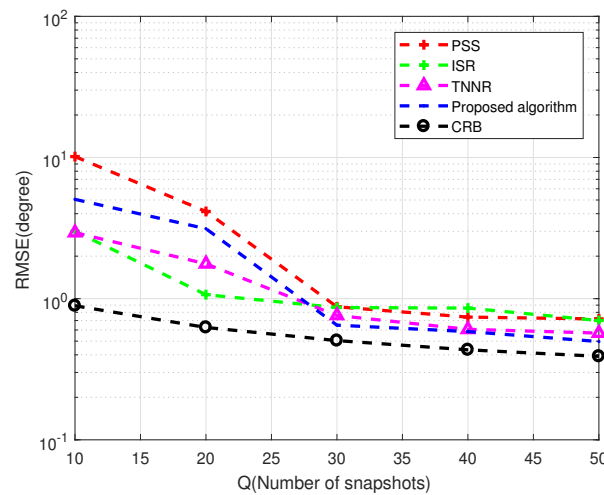


Figure 7. RMSE versus the number of snapshots for two uncorrelated targets.

In the fourth experiment, we evaluate the estimation performance of all tested algorithms for closely spaced targets. We assume two closely spaced targets are located at $(\theta_1, \varphi_1) = (10^\circ, 10^\circ)$ and $(\theta_2, \varphi_2) = (10^\circ + \Delta, 10^\circ + \Delta)$, where Δ varies from 2° to 10° . Figure 8 shows the RMSE versus the angular separation, where the number of snapshots is 30, and the SNR is fixed at 0 dB. As shown in Figure 8, all the test algorithms fail when the angular separation is less than or equal to 4° . The rationale is that the two closely spaced targets will cause the rank loss of the cross-correlation matrix between the sub-arrays, thereby reducing the angle estimation performance. When the angular separation is more significant than 4° , the proposed algorithm shows optimal estimation performance compared to the ISR, TNNR, and PSS algorithms. The reason is that the proposed algorithm obtains the largest virtual aperture by the enhanced matrix completion technique, thereby improving the estimation performance. However, the ISR algorithm only utilizes part of the consecutive lags, so the performance of this algorithm is lower than that of the proposed algorithm. Since the TNNR algorithm cannot complete the empty elements of the entire rows and columns, its performance is lower than that of the proposed algorithm. However, it has more virtual apertures than the ISR and PSS algorithms, so the estimation performance is better than these two algorithms. Since the PSS algorithm exploits the co-prime property to remove ambiguity and does not exploit virtual aperture expansion, it has the worst estimation performance compared to the proposed algorithm and the ISR algorithm.

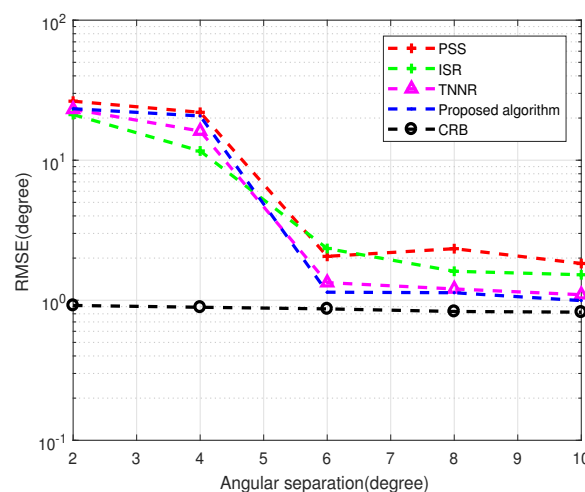


Figure 8. RMSE versus the angular separation for two uncorrelated targets.

In the fifth experiment, we evaluate the RMSE of the proposed algorithm and the TNNR algorithm versus the percentage of observations. We set that the received data of a 19×19 URA is \mathbf{X}_{out} , where the inter-element spacing is half a wavelength, and the SNR is set to 10 dB and $Q = 1$. Suppose two targets are located at $(\theta_1, \varphi_1) = (10^\circ, 10^\circ)$ and $(\theta_2, \varphi_2) = (30^\circ, 30^\circ)$, and the searching grid is set to 0.002 for $u \in [0, 1]$ and $v \in [0, 1]$. We randomly select [50%, 60%, 70%, 80%, 90%] elements of \mathbf{X}_{out} as available observation elements. Figure 9 shows RMSE versus the percentage of observations. It can be seen from Figure 9 that, when there are no missing elements in the entire row or column, the TNNR algorithm is slightly better than the proposed algorithm because the TNNR algorithm knows accurately the number of targets. However, the number of targets in practical problems is usually unknown, so the proposed algorithm has more practical application value.

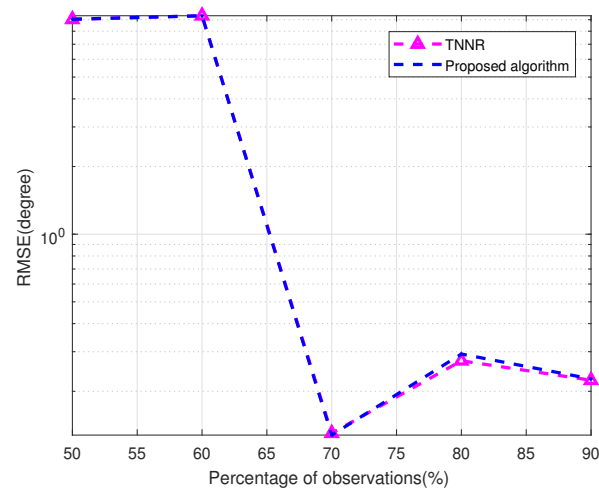


Figure 9. RMSE versus percentage of observation.

In the sixth experiment, we compare the RMSE and CPU time of all tested algorithms versus the number of $N = [3, 5, 7, 9]$, where M is fixed to 2. We assume two targets are located at $(\theta_1, \varphi_1) = (10^\circ, 10^\circ)$ and $(\theta_2, \varphi_2) = (30^\circ, 30^\circ)$, where the SNR is 0 dB, the number of snapshots is 50, and the searching grid is 0.0025 for $u \in [0, 1]$ and $v \in [0, 1]$. Figure 10 shows RMSE versus the total number of sensors. From Figure 10, as the number of sensors increases, the performance of all the tested algorithms increases. However, when the number of sensors is greater than 64, the performance of all the tested algorithms tends to be stable because the sampling grid limits further improvement in performance. Compared with the other tested algorithms, the proposed algorithm has the best performance because the proposed algorithm utilizes co-array interpolation initialization and enhanced matrix completion techniques. Figure 11 shows CPU time versus the total number of sensors, where the CPU time of all tested algorithms is based on the AMD R7-5800H processor.

From Figure 11, the CPU time of all the tested algorithms increases as the number of sensors increases. When the total number of sensors is less than 64, the CPU time of the proposed algorithm is higher than that of the PSS and TNNR algorithms and lower than that of the ISR algorithm. When the total number of sensors exceeds 64, the proposed algorithm has the highest CPU time due to the proposed algorithm's co-array interpolation and enhanced matrix completion. Furthermore, the proposed algorithm has better estimation accuracy than other tested algorithms under certain conditions. Therefore, the proposed algorithm has a better trade-off between computational complexity and estimation accuracy.

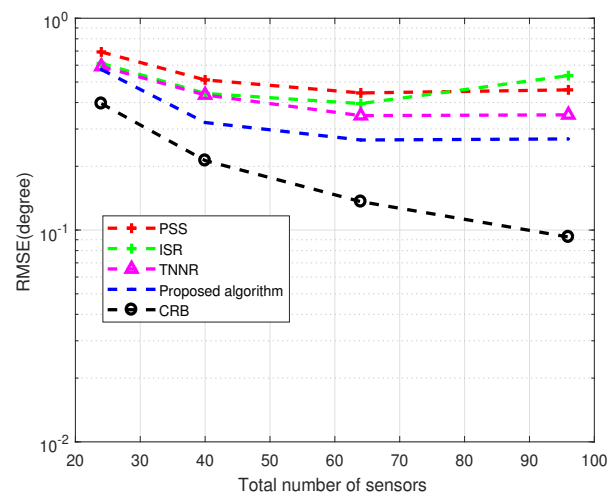


Figure 10. RMSE versus the total number of sensors.

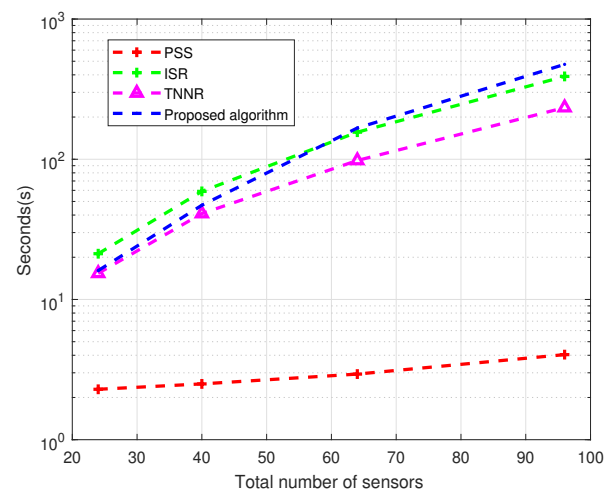


Figure 11. CPU time versus the total number of sensors.

4. Discussion

To exploit the virtual aperture of CPPA, we propose a sparsity-based, 2-D DOA estimation algorithm via enhanced matrix completion and co-array interpolation techniques. The first experiment in Section 3 shows that the proposed algorithm has the most significant DOF compared to the ISR, TNNR, and PSS algorithms. Therefore, the proposed algorithm can distinguish more targets than the ISR, TNNR, and PSS algorithms, consistent with the conclusions of [24].

In this study, we consider the estimation performance of the proposed method versus SNR and the number of snapshots. According to Figures 6 and 7, we find that the proposed algorithm has a better estimation performance than the ISR, TNNR, and PSS algorithms when the SNR is greater than 4 dB and the number of snapshots is greater than 20. The reason is that the proposed algorithm utilizes the virtual aperture expansion of CPPA and the sparse matrix recovery. In addition, when performing matrix completion, we utilize the shift-invariant structure of the virtual URA, thereby increasing the matrix completion accuracy, consistent with the ideas of [29]. The ISR algorithm performs better when the SNR is less than 4 dB and the number of snapshots is less than 20. The cause is reduced accuracy of the proposed enhanced matrix completion under low SNR and low snapshot conditions, resulting in a basis mismatch phenomenon when performing sparse matrix recovery, which reduces the estimation performance. Therefore, the DOA estimation performance of the proposed method is not only controlled by the virtual aperture, but also related to the SNR and the number of snapshots. From Figures 10 and 11, the proposed algorithm's estimated

performance and CPU time increase as the number of sensors increases. However, when the number of sensors exceeds 64, the proposed algorithm has a higher CPU time than the other tested algorithms. Therefore, the proposed algorithm is suitable for scenarios with a small number of sensors. Furthermore, the performances of the proposed algorithm, the ISR algorithm, the TNNR algorithm, and the PSS algorithm become relatively flat as the SNR and the number of snapshots and sensors increase. Due to the pre-defined sampling interval limiting further improvement in estimation performance, this is consistent with the conclusions of [17].

However, the proposed method also has certain limitations. When performing virtual aperture expansion, we do not consider the position and phase errors of the sensors, which are unavoidable in practical applications. If there are sensor position and phase errors, this will significantly reduce the matrix completion accuracy of the proposed algorithm, thereby reducing the estimation performance. Therefore, in future research, we need to consider the position and phase errors of the sensors and construct a matrix completion model with errors to further improve the robustness of matrix completion.

5. Conclusions

This paper proposes a complex-valued, sparse matrix-recovery-based 2-D DOA estimation algorithm for CPPA through enhanced matrix completion. First, we use the co-array interpolation technique to reconstruct two covariance matrices with the Toeplitz structure to initialize the holes of the SDCA, thereby constructing a virtual URA that does not contain missing elements of the entire rows and columns. Subsequently, an enhanced matrix completion model based on nuclear norm minimization is developed to complete all empty elements in the virtual URA, avoiding aperture loss. To reduce the computational burden of matrix completion, we develop a fast iterative algorithm based on the ADMM framework. Finally, we derive a complex-valued, sparse matrix-recovery algorithm based on the FISTA method to estimate the 2-D DOA of targets to reduce the computational complexity of the traditional vector-form, sparse recovery algorithm. The simulation results demonstrate the superior performance of the proposed algorithm.

Author Contributions: Y.Z. designed the experiments; D.L. performed the experiments and wrote the paper; T.Z. revised the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No.B18039), partly by the Fundamental Research Funds for the Central Universities, and partly by the Innovation Fund of Xidian University.

Data Availability Statement: Not applicable

Acknowledgments: The authors would like to thank the reviewers for their valuable suggestions on improving this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

In Appendix A, we prove the derivation of (51). According to (48) and (49), $\nabla H(\mathbf{X})$ can be expressed as

$$\begin{aligned} \nabla H(\mathbf{X}) &= \frac{1}{2} \nabla_{\mathbf{X}} \operatorname{tr} \left\{ \left(\hat{\mathbf{Y}} - \tilde{\mathbf{A}} \mathbf{X} \tilde{\mathbf{B}}^T \right)^H \left(\hat{\mathbf{Y}} - \tilde{\mathbf{A}} \mathbf{X} \tilde{\mathbf{B}}^T \right) \right\} \\ &= \frac{1}{2} \nabla_{\mathbf{X}} \left\{ \operatorname{tr} \left(\hat{\mathbf{Y}}^H \hat{\mathbf{Y}} \right) - \operatorname{tr} \left(\tilde{\mathbf{B}}^T \hat{\mathbf{Y}}^H \tilde{\mathbf{A}} \mathbf{X} \right) - \right. \\ &\quad \left. \operatorname{tr} \left(\tilde{\mathbf{A}}^H \hat{\mathbf{Y}} \tilde{\mathbf{B}}^* \mathbf{X}^H \right) + \operatorname{tr} \left(\tilde{\mathbf{B}}^T \tilde{\mathbf{B}}^* \mathbf{X}^H \tilde{\mathbf{A}}^H \tilde{\mathbf{A}} \mathbf{X} \right) \right\} \\ &= \tilde{\mathbf{A}}^H \left(\tilde{\mathbf{A}} \mathbf{X} \tilde{\mathbf{B}}^T - \hat{\mathbf{Y}} \right) \tilde{\mathbf{B}}^* \end{aligned} \quad (\text{A1})$$

Then, we substitute (A1) into (49), and, after some algebraic processing, we obtain

$$\begin{aligned}
 Q_L(\mathbf{X}, \mathbf{P}) &= \text{tr}(\nabla H(\mathbf{P})^H(\mathbf{X} - \mathbf{P})) + \frac{L_f}{2} \text{tr}((\mathbf{X} - \mathbf{P})^H) \\
 &+ \rho F(\mathbf{X}) + H(\mathbf{X}_1) \\
 &= \frac{L_f}{2} \text{tr}\left(\left(\mathbf{X} - \left(\mathbf{P} - \frac{1}{L_f} \nabla H(\mathbf{P})\right)\right)^H (\mathbf{X} - \mathbf{P})\right) \\
 &+ \rho F(\mathbf{X}) + H(\mathbf{P}) \\
 &= \frac{L_f}{2} \|\mathbf{X} - U(\mathbf{P})\|_F^2 + \rho F(\mathbf{X}) + C(\mathbf{P})
 \end{aligned} \tag{A2}$$

where $C(\mathbf{P})$ is a constant matrix of \mathbf{P} and $U(\mathbf{P})$ can be given as

$$U(\mathbf{P}) = \mathbf{P} - \frac{1}{L_f} \tilde{\mathbf{A}}^H (\tilde{\mathbf{A}} \mathbf{X} \tilde{\mathbf{B}}^T - \hat{\mathbf{Y}}) \tilde{\mathbf{B}}^* \tag{A3}$$

this completes the proof.

References

- Krim, H.; Viberg, M. Two decades of array signal processing research: The parametric approach. *IEEE Signal Process. Mag.* **1996**, *13*, 67–94. [\[CrossRef\]](#)
- Choi, J.; Park, J.; Yeom, D. High angular resolution estimation methods for vehicle FMCW radar. In Proceedings of the 2011 IEEE CIE International Conference on Radar, Chengdu, China, 24–27 October 2011; Volume 2, pp. 1868–1871.
- Gu, Y.; Goodman, N.A. Information-Theoretic Compressive Sensing Kernel Optimization and Bayesian Cramér–Rao Bound for Time Delay Estimation. *IEEE Trans. Signal Process.* **2017**, *65*, 4525–4537. [\[CrossRef\]](#)
- Schmidt, R. Multiple emitter location and signal parameter estimation. *IEEE Trans. Antennas Propag.* **1986**, *34*, 276–280. [\[CrossRef\]](#)
- Roy, R.; Kailath, T. ESPRIT-estimation of signal parameters via rotational invariance techniques. *IEEE Trans. Acoust. Speech Signal Process.* **1989**, *37*, 984–995. [\[CrossRef\]](#)
- Zoltowski, M.; Silverstein, S.; Mathews, C. Beamspace Root-MUSIC for minimum redundancy linear arrays. *IEEE Trans. Signal Process.* **1993**, *41*, 2502–2507. [\[CrossRef\]](#)
- Zoltowski, M.; Haardt, M.; Mathews, C. Closed-form 2-D angle estimation with rectangular arrays in element space or beamspace via unitary ESPRIT. *IEEE Trans. Signal Process.* **1996**, *44*, 316–328. [\[CrossRef\]](#)
- Xu, B.; Zhao, Y. Transmit beamspace-based DOD and DOA estimation method for bistatic MIMO radar. *Signal Process.* **2019**, *157*, 88–96. [\[CrossRef\]](#)
- Liu, D.; Zhao, Y.; Cao, C.; Pang, X. A novel reduced-dimensional beamspace unitary ESPRIT algorithm for monostatic MIMO radar. *Digit. Signal Process.* **2021**, *114*, 103027. [\[CrossRef\]](#)
- Vaidyanathan, P.P.; Pal, P. Sparse sensing with coprime arrays. In Proceedings of the 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 7–10 November 2010; pp. 1405–1409.
- Vaidyanathan, P.P.; Pal, P. Sparse Sensing With Co-Prime Samplers and Arrays. *IEEE Trans. Signal Process.* **2011**, *59*, 573–586. [\[CrossRef\]](#)
- Liu, C.L.; Vaidyanathan, P.P. Remarks on the Spatial Smoothing Step in Coarray MUSIC. *IEEE Signal Process. Lett.* **2015**, *22*, 1438–1442. [\[CrossRef\]](#)
- Zhang, Y.D.; Amin, M.G.; Himed, B. Sparsity-based DOA estimation using co-prime arrays. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 3967–3971.
- Liu, C.L.; Vaidyanathan, P.P.; Pal, P. Coprime coarray interpolation for DOA estimation via nuclear norm minimization. In Proceedings of the 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, Canada, 22–25 May 2016; pp. 2639–2642.
- Fan, X.; Zhou, C.; Gu, Y.; Shi, Z. Toeplitz Matrix Reconstruction of Interpolated Coprime Virtual Array for DOA Estimation. In Proceedings of the 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, NSW, Australia, 4–7 June 2017; pp. 1–5.
- Zheng, Y.; Guo, M.; Liu, L. DOA Estimation Using Coarray Interpolation Algorithm Via Nuclear Norm Optimization for Coprime MIMO Radar. In Proceedings of the 2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM), Hangzhou, China, 8–11 June 2020; pp. 1–5.
- Zhou, C.; Gu, Y.; Fan, X.; Shi, Z.; Mao, G.; Zhang, Y.D. Direction-of-Arrival Estimation for Coprime Array via Virtual Array Interpolation. *IEEE Trans. Signal Process.* **2018**, *66*, 5956–5971. [\[CrossRef\]](#)
- Zhou, C.; Gu, Y.; Shi, Z.; Zhang, Y.D. Off-Grid Direction-of-Arrival Estimation Using Coprime Array Interpolation. *IEEE Signal Process. Lett.* **2018**, *25*, 1710–1714. [\[CrossRef\]](#)
- Yadav, S.K.; George, N.V. Fast Direction-of-Arrival Estimation via Coarray Interpolation Based on Truncated Nuclear Norm Regularization. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 1522–1526. [\[CrossRef\]](#)

20. Fan, Q.; Liu, Y.; Yang, T.; Peng, H. Fast and Accurate Spectrum Estimation via Virtual Coarray Interpolation Based on Truncated Nuclear Norm Regularization. *IEEE Signal Process. Lett.* **2022**, *29*, 169–173. [[CrossRef](#)]
21. Cheng, Z.; Zhao, Y.; Li, H.; Shui, P. Two-dimensional DOA estimation algorithm with co-prime array via sparse representation. *Electron. Lett.* **2015**, *51*, 2084–2086. [[CrossRef](#)]
22. Xu, H.; Zhang, Y.; Ba, B.; Wang, D.; Li, X. Two-Dimensional Direction-of-Arrival Fast Estimation of Multiple Signals with Matrix Completion Theory in Coprime Planar Array. *Sensors* **2018**, *18*, 1741. [[CrossRef](#)]
23. Wu, Q.; Sun, F.; Lan, P.; Ding, G.; Zhang, X. Two-Dimensional Direction-of-Arrival Estimation for Co-Prime Planar Arrays: A Partial Spectral Search Approach. *IEEE Sens. J.* **2016**, *16*, 5660–5670. [[CrossRef](#)]
24. Shi, J.; Hu, G.; Zhang, X.; Sun, F.; Zhou, H. Sparsity-Based Two-Dimensional DOA Estimation for Coprime Array: From Sum–Difference Coarray Viewpoint. *IEEE Trans. Signal Process.* **2017**, *65*, 5591–5604. [[CrossRef](#)]
25. Nguyen, L.T.; Kim, J.; Shim, B. Low-Rank Matrix Completion: A Contemporary Survey. *IEEE Access* **2019**, *7*, 94215–94237. [[CrossRef](#)]
26. He, B.; Tao, M.; Xu, M.; Yuan, X. An alternating direction-based contraction method for linearly constrained separable convex programming problems. *Optimization* **2013**, *62*, 573–596. [[CrossRef](#)]
27. Beck, A.; Teboulle, M. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [[CrossRef](#)]
28. Vandenberghe, L.; Boyd, S.; Wu, S.P. Determinant Maximization with Linear Matrix Inequality Constraints. *Siam J. Matrix Anal. Appl.* **1998**, *19*, 499–533. [[CrossRef](#)]
29. Chen, Y.; Chi, Y. Robust Spectral Compressed Sensing via Structured Matrix Completion. *IEEE Trans. Inf. Theory* **2014**, *60*, 6576–6601. [[CrossRef](#)]
30. Cai, J.F.; Candès, E.J.; Shen, Z. A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM J. Optim.* **2010**, *20*, 1956–1982. [[CrossRef](#)]
31. Duchi, J.; Shalev-Shwartz, S.; Singer, Y.; Chandra, T. Efficient projections onto the l1-ball for learning in high dimensions. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; ACM: New York, NY, USA, 2008; pp. 272–279.
32. Wimalajeewa, T.; Eldar, Y.; Varshney, P. Recovery of Sparse Matrices via Matrix Sketching. *Comput. Sci.* **2013**.
33. Wright, S.J.; Nowak, R.D.; Figueiredo, M.A.T. Sparse Reconstruction by Separable Approximation. *IEEE Trans. Signal Process.* **2009**, *57*, 2479–2493. [[CrossRef](#)]