



# Synthesis of sparse planar arrays with multiple patterns: An automatic coordinate pairing approach

Yangjingzhi Zhuang, Xuejing Zhang\*, Zishu He

*School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China*

## ARTICLE INFO

### Article history:

Received 8 March 2023

Revised 24 April 2023

Accepted 19 June 2023

Available online 25 June 2023

### Keywords:

Antenna pattern synthesis

Multiple patterns

Sparse planar array

## ABSTRACT

An innovative algorithm to synthesize a non-uniform sparse planar array with as few elements as possible is proposed in this paper. The algorithm is aimed at multiple different beampatterns, and can directly estimate the exact positions of the planar array elements, avoiding the problem of coordinate pairing between X-axis and Y-axis. To achieve this, the proposed method utilizes the rotational invariance caused by the translation invariant structure of the steering vector. It derives the array element position from the data matrix based on unitary ESPRIT and employs a low-rank approximation operation to construct a matrix containing the position information of the reconstructed sparse array elements. Automatic pairing is then realized by assigning the real and imaginary parts of the matrix eigenvalues to the abscissa and ordinate of elements one-to-one. The paper presents two different simulation scenarios to demonstrate the performance of the proposed algorithm, with the results showing a reduction in the number of array elements by 20%-30% while ensuring low reconstruction error, low time, and space complexity. Overall, the proposed algorithm provides an effective solution for synthesizing non-uniform sparse planar arrays with fewer elements, which can significantly reduce costs and complexity in various applications.

© 2023 Elsevier B.V. All rights reserved.

## 1. Introduction

Synthesizing the desired beampattern with fewer array elements is a subject of ongoing research, as highlighted in recent studies [1–3]. Sparse arrays offer several advantages over traditional full arrays. For instance, for a given number of elements, a sparse array has a larger array aperture, which enhances the accuracy and resolution of the direction of arrival algorithm [4,5]. Additionally, a sparse array requires fewer elements to achieve the same array aperture, leading to smaller-scale receiving and signal processing systems. Hence, the cost can be greatly reduced by applying sparse array [6]. Sparse arrays also reduce mutual coupling [7] between array elements, thereby weakening its impact on the performance. Given these benefits, exploring the applications and designs of sparse arrays is crucial [8].

Generally speaking, deleting or closing some elements from an equally spaced array is a typical way of forming a sparse array, where the spacing between two arbitrary elements is an integral multiple of a fixed space. Stochastic optimization algorithm such as genetic algorithm (GA) [9], simulated annealing (SA) [10], particle

swarm optimization (PSO) [11–13] and differential evolution algorithm (DE) [14–16] are commonly used for this purpose. However, these algorithms have high computational complexity, especially for large arrays, and may get stuck in local optima. In [17,18], Keizer proposed an iterative fast Fourier transform algorithm, which removes some radiating elements to obtain a new sparse array by using inverse discrete Fourier transform relationship between the array factor and the element excitations. This method significantly reduces computational complexity, but its effectiveness depends heavily on the choice of the starting point. With the development of convex optimization, it has been utilized to synthesize sparse arrays; see the work in [19,20]. Notably, a convex optimization algorithm proposed in [21] for the synthesis of sparse arrays radiating focused or shaped beampattern cannot guarantee optimality of the final result.

In the above methods, the candidate antennas are all restricted to the grid, thus limiting the degree of freedom for the arrangement of array elements. Conversely, a flexible sparse linear array reconstruction technique based on the matrix pencil method (MPM) was proposed in [22,23], reconstructed array antennas are not limited to grids. On this basis, Liu et al. proposed the forward-backward matrix pencil method (FBMPM) by improving the MPM in [24] and extended MPM and FBMPM to multiple-pattern scenario in [25]. MPM performs sparse array synthesis by constructing Hankel matrices and utilizing low-rank approximation. Although

\* Corresponding author.

E-mail addresses: [zhuangyangjingzhi@163.com](mailto:zhuangyangjingzhi@163.com) (Y. Zhuang), [zhangxuejing@uestc.edu.cn](mailto:zhangxuejing@uestc.edu.cn), [xjzhang7@163.com](mailto:xjzhang7@163.com) (X. Zhang), [\(Z. He\)](mailto:zshe@uestc.edu.cn).

presenting high performance, the application of MPM is limited to linear arrays. Based on the matrix enhancement and matrix pencil (MEMP) in [26], Gu et al. introduced a method to synthesize sparse planar arrays in [27]. Extending to multiple patterns, the generalized matrix enhancement and matrix pencil (GMEPM) was presented in [28]. However, according to the principle of MPM, its computational complexity significantly increases as the number of patterns increases. In addition, there is another type of method for synthesizing sparse arrays based on the Bayesian compression sensing (BCS) technique, as proposed in [29–33], and the candidate antennas may not limited to grids. BCS obtains both the optimal positions and excitations of a new sparse array through a relevance vector machine. It should be noted that an additional coordinate pairing step is required for the above-mentioned methods when synthesizing a planar array. Also, the coordinate pairing operation increases the computational complexity and may lead to performance degradation due to some potential pairing errors.

In this paper, we propose a method for synthesis of multiple-pattern sparse planar arrays with automatic coordinate pairing. Different from MPM methods that rely on constructing Hankel matrices, the proposed method arranges the sampled data of desired radiation pattern in a regular way. In this method, a special unitary transformation is firstly performed on the uniformly sampled steering vectors to obtain a real-valued vector, and then a uniformly sampled data matrix is obtained. Next, based on unitary ESPRIT[34], we exploit the rotation invariance property arising from the translation invariance structure of the steering vector to derive the array element positions from the data matrix. In order to reduce the number of array elements, a matrix containing position information of the reconstructed sparse array elements is constructed by low-rank approximation. And real and imaginary parts of the eigenvalues of the matrix are in one-to-one correspondence with the abscissa and ordinate of elements positions. This way, we can synthesize sparse planar arrays without coordinate pairing between X-axis and Y-axis. The algorithm can directly estimate exact positions of array elements, avoiding the coordinate pairing problem. In addition, the obtained new array can synthesize multiple different beampatterns that fit the requirements, not limited to a single pattern. Moreover, the proposed method has low computational and space complexity. Representative simulations are conducted in different scenarios to validate the feasibility and effectiveness of the algorithm.

The rest of this paper is organized as follows. The problem formulation and unitary transformation are presented in Section 2. The proposed method is analyzed in Section 3. In Section 4, numerical results are provided to verify the performance of the proposed method. Finally, conclusions are drawn in Section 5.

## 2. Preliminaries

### 2.1. Problem formulation

We consider a uniform rectangular array (URA) with  $G = G_r \times G_c$  antennas distributed in the  $x - y$  plane. For simplicity, we assume that all antennas are isotropic. Considering  $P$  desired patterns, we can express the  $p$ th beampattern as follows:

$$F^{(p)}(u, v) = \sum_{g=1}^G R_g^{(p)} e^{j\beta(x_g u + y_g v)}, \quad p = 1, 2, \dots, P \quad (1)$$

where  $R_g^{(p)}$  represents the complex excitation coefficient for the  $p$ th pattern of the  $g$ th element with coordinate  $(x_g, y_g)$ ,  $x_g$  and  $y_g$  stand for the abscissa and ordinate of the  $g$ th element, respectively,  $g = 1, 2, \dots, G$ .  $\beta = 2\pi/\lambda$ ,  $\lambda$  is the wavelength.  $j = \sqrt{-1}$  is the imaginary unit, and  $u = \sin\theta\cos\varphi$ ,  $v = \sin\theta\sin\varphi$ , where  $\theta$  is the angle with the  $z$  axis, and  $\varphi$  is azimuth. The desired patterns can

be obtained through optimization algorithms such as GA, DE, etc. The purpose of sparse array synthesis is to construct a new array with minimum number of antennas. And the sparse array can keep the synthesized patterns and the specified desired patterns  $F(u, v)$  within an acceptable tolerance  $\varepsilon$ . The problem can be described as (2), shown at the top of the next page,

$$\begin{cases} \min_{R_m^{(p)}, x'_m, y'_m} & M \\ \text{s.t.} & \sum_{p=1}^P \int_{-1}^1 \int_{-1}^1 \left| F^{(p)}(u, v) - \sum_{m=1}^M R_m^{(p)}' \exp[jk(x'_m u + y'_m v)] \right|^2 dudv \leq \varepsilon \end{cases} \quad (2)$$

where  $M$  is the total number of reconstructed array antennas, and  $R_m^{(p)'}$  denotes the new complex excitation coefficient for the  $p$ th pattern of the  $m$ th new array element at the coordinate of  $(x'_m, y'_m)$ ,  $m = 1, 2, \dots, M$ .

### 2.2. Unitary transformation

Given a vector  $\mathbf{b} \in \mathbb{C}^L$ , then  $\mathbf{b}$  is conjugate centrosymmetry if it satisfies:

$$\Pi_L \mathbf{b} = \mathbf{b}^* \quad (3)$$

where

$$\Pi_L = \begin{bmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \dots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{L \times L} \quad (4)$$

Since the inner product of any two conjugate centrosymmetric vectors is real-valued, it is not difficult to find that any conjugate centrosymmetric vector multiply by a matrix from the left side returns a real-valued vector, provided that the matrix has conjugate centrosymmetric row vectors. One simple candidate for the above matrices is:

$$\mathbf{Q}_L = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_K & j\mathbf{I}_K \\ \mathbf{I}_K & -j\mathbf{I}_K \end{bmatrix}, \quad L = 2K \quad (5)$$

$$\mathbf{Q}_L = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_K & \mathbf{0}_K & j\mathbf{I}_K \\ \mathbf{0}_K^T & \sqrt{2} & \mathbf{0}_K^T \\ \mathbf{I}_K & \mathbf{0}_K & -j\mathbf{I}_K \end{bmatrix}, \quad L = 2K + 1 \quad (6)$$

where  $\mathbf{I}_K$  represents the  $K \times K$  identity matrix, and  $\mathbf{0}_K$  is a  $K$ -dimensional zero vector. Note that  $\mathbf{Q}_L$  is an  $L \times L$  unitary matrix.

## 3. Reconstructed planar array algorithm with automatic coordinate pairing

### 3.1. Array model

We consider the array as described in Section 2.1. One can see that both  $u$  and  $v$  are defined in the domain  $[-1, 1]$ . Sampling  $u$  and  $v$  uniformly, we obtain:

$$u_{n_1} = n_1 \Delta = \frac{n_1}{N}, \quad n_1 = -N, \dots, 0, \dots, N \quad (7)$$

$$v_{n_2} = n_2 \Delta = \frac{n_2}{N}, \quad n_2 = -N, \dots, 0, \dots, N \quad (8)$$

where the number of samples on both  $u$  and  $v$  is  $L = (2N + 1)$ , the total number of sampling points is  $L^2 = (2N + 1)^2$  and  $\Delta = \frac{1}{N}$  is the sampling interval. Due to the Nyquist sampling theorem, the condition that  $\Delta \leq \lambda/\min\{2x_{\max}, 2y_{\max}\}$  must be satisfied,

where  $x_{\max} = \max\{x_1, \dots, x_G\}$ ,  $y_{\max} = \max\{y_1, \dots, y_G\}$ . For example, for an array having  $G = G_r \times G_c$  elements with a uniform spacing of  $\lambda/2$ , the sampling interval  $\Delta$  should satisfy  $\Delta \leq \min\{1/(G_r - 1), 1/(G_c - 1)\}$ , which indicates that  $N \geq \max\{G_r - 1, G_c - 1\}$ .

Before proceeding, for the  $g$ th element we define:

$$\mathbf{b}(x_g) = [e^{-j\beta x_g(-N)\Delta}, e^{-j\beta x_g(-N+1)\Delta}, \dots, e^{-j\beta x_g N\Delta}]^T \quad (9)$$

$$\mathbf{b}(y_g) = [e^{-j\beta y_g(-N)\Delta}, e^{-j\beta y_g(-N+1)\Delta}, \dots, e^{-j\beta y_g N\Delta}]^T \quad (10)$$

$$\mathbf{A}_g = \mathbf{b}(x_g)\mathbf{b}^T(y_g) \in \mathbb{C}^{L \times L} \quad (11)$$

And the matrix  $\mathbf{Z} = [\text{vec}(\mathbf{A}_1), \text{vec}(\mathbf{A}_2), \dots, \text{vec}(\mathbf{A}_G)]$  is constructed, where  $\text{vec}(\cdot)$  is the vectorization operation. It is worth noting that each row of the matrix  $\mathbf{Z}$  is the steering vector of the array, which can be expressed as

$$\mathbf{Z} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_{L^2}^T \end{bmatrix} \quad (12)$$

where

$$\mathbf{a}_i = [e^{-j\beta(x_1 u_i + y_1 v_i)}, e^{-j\beta(x_2 u_i + y_2 v_i)}, \dots, e^{-j\beta(x_G u_i + y_G v_i)}]^T \quad (13)$$

is the steering vector corresponding to the  $i$ th sampling point, and  $i = 1, 2, \dots, L^2$ .

Since  $L = (2N + 1)$  is odd, the form in (6) is used and

$$\begin{aligned} \mathbf{d}(x_g) &= \mathbf{Q}_L^H \mathbf{b}(x_g) \\ &= \sqrt{2} \times [\cos(Nx_g), \dots, \cos(x_g), 1/\sqrt{2}, \\ &\quad -\sin(x_g), \dots, -\sin(Nx_g)]^T \end{aligned} \quad (14)$$

Similarly, we can get the real-valued vector  $\mathbf{d}(y_g) = \mathbf{Q}_L^H \mathbf{b}(y_g)$ . And a matrix about the  $g$ th element can be represented as follows:

$$\begin{aligned} \mathbf{D}_g &= \mathbf{Q}_L^H \mathbf{A}_g \mathbf{Q}_L^* \\ &= \mathbf{Q}_L^H \mathbf{b}(x_g) \mathbf{b}^T(y_g) \mathbf{Q}_L^* \\ &= \mathbf{d}(x_g) \mathbf{d}^T(y_g) \end{aligned} \quad (15)$$

and a real-valued matrix is defined as  $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_G]$ .

### 3.2. Rearranging elements with automatic coordinate pairing

We know that the desired beampatterns can be obtained by adjusting the weight vector  $\mathbf{w}$ . For a given weight vector  $\mathbf{w}_p$ , the beampattern may be expressed as

$$\mathbf{y}_p = \mathbf{Z} \mathbf{w}_p \quad (16)$$

Through different weight vectors, the beampatterns pointing in different directions are

$$\mathbf{Y} = \mathbf{Z} \mathbf{W} \quad (17)$$

where  $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_P]$  and  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_P]$ . And  $\mathbf{y}_p$  is obtained by the  $p$ th weight vector  $\mathbf{w}_p$ , as expressed in (16).

Next, the method for estimating the positions of new sparse array elements is introduced. Based on unitary ESPRIT in [34], a real-valued  $L^2 \times 2P$  matrix is defined as

$$\tilde{\mathbf{Y}} = [\text{Re}\{\tilde{\mathbf{Y}}\}, \text{Im}\{\tilde{\mathbf{Y}}\}] \quad (18)$$

where  $\tilde{\mathbf{Y}} = (\mathbf{Q}_L^H \otimes \mathbf{Q}_L^H) \mathbf{Y}$  and  $\otimes$  denotes the Kronecker matrix product. Splitting the real and imaginary parts of  $\tilde{\mathbf{Y}}$  to constitute  $\tilde{\mathbf{Y}}$  can increase the length of valid data, thereby improving the estimation accuracy about the new array elements positions of the proposed algorithm. Moreover, the real-value matrix  $\tilde{\mathbf{Y}}$  that needs to be processed finally reduces the computational complexity of subsequent procedure, such as SVD operations, etc.

Next, the SVD of the matrix  $\tilde{\mathbf{Y}}$  is carried out as

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{U}} \tilde{\Sigma} \tilde{\mathbf{V}}^H \quad (19)$$

where  $\tilde{\mathbf{U}} \in \mathbb{R}^{L^2 \times L^2}$  and  $\tilde{\mathbf{V}} \in \mathbb{R}^{2P \times 2P}$  are unitary matrices, and

$$\tilde{\Sigma} = \begin{bmatrix} \tilde{\Sigma}_1 \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{L^2 \times 2P} \quad (20)$$

where  $\tilde{\Sigma}_1 = \text{diag}(\tilde{\sigma}_1, \tilde{\sigma}_2, \dots, \tilde{\sigma}_{\tilde{S}})$ , and  $\tilde{\sigma}_{\tilde{S}}$  is the ordered  $\tilde{S}$ th singular value of  $\tilde{\mathbf{Y}}$ , and  $\tilde{S} = \min\{2P, L^2\}$ . In order to contain the information of the initial beampatterns completely, we set  $L^2 > 2P$  here, so we have  $\tilde{S} = 2P$ .

The vector  $\mathbf{b}(x_g)$  in (9) satisfies the invariance relation

$$e^{jx_g} \mathbf{J}_1 \mathbf{b}(x_g) = \mathbf{J}_2 \mathbf{b}(x_g) \quad (21)$$

where  $\mathbf{J}_1$  and  $\mathbf{J}_2$  are the  $(L - 1) \times L$  selection matrices

$$\mathbf{J}_1 = [\mathbf{I}_{L-1} \quad \mathbf{0}] \in \mathbb{R}^{(L-1) \times L} \quad (22)$$

$$\mathbf{J}_2 = [\mathbf{0} \quad \mathbf{I}_{L-1}] \in \mathbb{R}^{(L-1) \times L} \quad (23)$$

where  $\mathbf{I}_{L-1}$  is an identity matrix of size  $(L - 1) \times (L - 1)$ . And multiplying an  $L$ -dimension vector with the  $\mathbf{J}_1$  and  $\mathbf{J}_2$  selects the first and last  $L - 1$  components of the vector, respectively. Since  $\mathbf{Q}_L$  is unitary, it follows:

$$e^{jx_g} \mathbf{J}_1 \mathbf{Q}_L \mathbf{Q}_L^H \mathbf{b}(x_g) = \mathbf{J}_2 \mathbf{Q}_L \mathbf{Q}_L^H \mathbf{b}(x_g) \quad (24)$$

According to (14), we establish an invariance relationship of  $\mathbf{d}(x_g)$

$$e^{jx_g} \mathbf{J}_1 \mathbf{Q}_L \mathbf{d}(x_g) = \mathbf{J}_2 \mathbf{Q}_L \mathbf{d}(x_g) \quad (25)$$

Multiplying  $\mathbf{Q}_{L-1}^H$ , an equation can be acquired:

$$e^{jx_g} \mathbf{Q}_{L-1}^H \mathbf{J}_1 \mathbf{Q}_L \mathbf{d}(x_g) = \mathbf{Q}_{L-1}^H \mathbf{J}_2 \mathbf{Q}_L \mathbf{d}(x_g) \quad (26)$$

Note that  $\mathbf{J}_1$  and  $\mathbf{J}_2$  satisfy  $\mathbf{J}_1 \mathbf{J}_2 \mathbf{Q}_L = \mathbf{J}_2 \mathbf{J}_1 \mathbf{Q}_L = \mathbf{J}_1$ . As a consequence,

$$\begin{aligned} \mathbf{Q}_{L-1}^H \mathbf{J}_2 \mathbf{Q}_L &= \mathbf{Q}_{L-1}^H \mathbf{J}_1 \mathbf{J}_2 \mathbf{Q}_L \mathbf{d}(x_g) \\ &= \mathbf{Q}_{L-1}^T \mathbf{J}_1 \mathbf{Q}_L^* \\ &= (\mathbf{Q}_{L-1}^H \mathbf{J}_1 \mathbf{Q}_L)^* \end{aligned} \quad (27)$$

where we have exploited the fact that  $\mathbf{Q}_L \mathbf{Q}_L^* = \mathbf{I}_L$  and  $\mathbf{Q}_L^* \mathbf{Q}_L = \mathbf{I}_L$  for any  $L$ . Then, substituting (27) into (26), we get

$$e^{jx_g} (\mathbf{Q}_{L-1}^H \mathbf{J}_2 \mathbf{Q}_L)^* \mathbf{d}(x_g) = \mathbf{Q}_{L-1}^H \mathbf{J}_2 \mathbf{Q}_L \mathbf{d}(x_g) \quad (28)$$

Let  $\mathbf{K}_1$  and  $\mathbf{K}_2$  be the real and imaginary parts of  $\mathbf{Q}_{L-1}^H \mathbf{J}_2 \mathbf{Q}_L$ , as follows:

$$\mathbf{K}_1 = \text{Re}\{\mathbf{Q}_{L-1}^H \mathbf{J}_2 \mathbf{Q}_L\} \quad (29)$$

$$\mathbf{K}_2 = \text{Im}\{\mathbf{Q}_{L-1}^H \mathbf{J}_2 \mathbf{Q}_L\} \quad (30)$$

$\mathbf{K}_1$  and  $\mathbf{K}_2$  are real-valued  $(L - 1) \times L$  matrices. With these definitions, Eq. (26) can be expressed as

$$e^{j\frac{x_g}{2}} (\mathbf{K}_1 - j\mathbf{K}_2) \mathbf{d}(x_g) = e^{-j\frac{x_g}{2}} (\mathbf{K}_1 + j\mathbf{K}_2) \mathbf{d}(x_g) \quad (31)$$

Rearranging, the above equation becomes

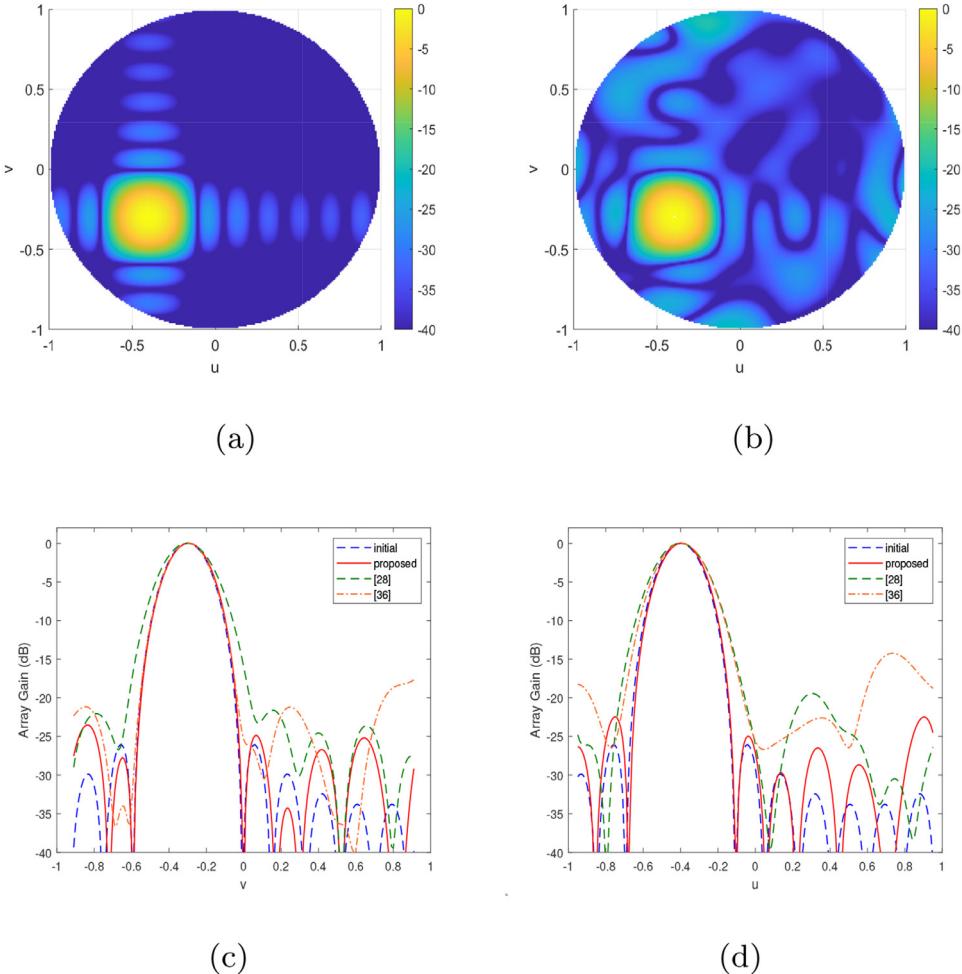
$$(e^{j\frac{x_g}{2}} - e^{-j\frac{x_g}{2}}) \mathbf{K}_1 \mathbf{d}(x_g) = j(e^{j\frac{x_g}{2}} + e^{-j\frac{x_g}{2}}) \mathbf{K}_2 \mathbf{d}(x_g) \quad (32)$$

Using Euler's formula, an invariance relationship is obtained as

$$\tan\left(\frac{x_g}{2}\right) \mathbf{K}_1 \mathbf{d}(x_g) = \mathbf{K}_2 \mathbf{d}(x_g) \quad (33)$$

Simply extended to the URA, the invariance relationship can be expressed as

$$\tan\left(\frac{x_g}{2}\right) \mathbf{K}_1 \mathbf{D}_g = \mathbf{K}_2 \mathbf{D}_g \quad (34)$$



**Fig. 1.** Comparison of initial and reconstructed patterns at  $u = -0.4$ ,  $v = -0.3$ . (a) Top view of initial patterns. (b) Top view of reconstructed patterns. (c) Slice diagram at  $u = -0.4$ . (d) Slice diagram at  $v = -0.3$ .

Using the vectorization operation  $\text{vec}(\cdot)$ , we find that the  $L^2$ -dimension real-valued vector  $\text{vec}(\mathbf{D}_g)$  satisfies

$$\tan\left(\frac{x_g}{2}\right)\mathbf{K}_{x1}\text{vec}(\mathbf{D}_g) = \mathbf{K}_{x2}\text{vec}(\mathbf{D}_g) \quad (35)$$

where  $\mathbf{K}_{x1} = \mathbf{I}_L \otimes \mathbf{K}_1$  and  $\mathbf{K}_{x2} = \mathbf{I}_L \otimes \mathbf{K}_2$  are the  $(L-1)L \times L^2$  matrices, and the following equation has been exploited from (34) to (35):

$$\text{vec}(\mathbf{BCH}) = (\mathbf{H}^T \otimes \mathbf{B})\text{vec}(\mathbf{C}) \quad (36)$$

where  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{H}$  are any matrices with appropriate dimensions. Similarly, the real-valued vector  $\mathbf{d}(y_g)$  satisfies

$$\tan\left(\frac{y_g}{2}\right)\mathbf{K}_3\mathbf{d}(y_g) = \mathbf{K}_4\mathbf{d}(y_g) \quad (37)$$

where  $\mathbf{K}_3$  and  $\mathbf{K}_4$  are defined by (29) and (30), i.e.  $\mathbf{K}_3$  and  $\mathbf{K}_4$  are the same as  $\mathbf{K}_1$  and  $\mathbf{K}_2$  due to the same sampling of  $u$  and  $v$ . It follows that

$$\tan\left(\frac{y_g}{2}\right)\mathbf{D}_g\mathbf{K}_3^T = \mathbf{D}_g\mathbf{K}_4^T \quad (38)$$

Similarly, we find that  $\mathbf{d}_g$  satisfies

$$\tan\left(\frac{y_g}{2}\right)\mathbf{K}_{y1}\text{vec}(\mathbf{D}_g) = \mathbf{K}_{y2}\text{vec}(\mathbf{D}_g) \quad (39)$$

where  $\mathbf{K}_{y1} = \mathbf{K}_3 \otimes \mathbf{I}_L$  and  $\mathbf{K}_{y2} = \mathbf{K}_4 \otimes \mathbf{I}_L$  are the  $L(L-1) \times L^2$  matrices.

Without loss of generality, we first assume that the new sparse array has  $M$  elements. For  $M < G$  elements, we define the  $L^2 \times M$  real-valued matrix as  $\check{\mathbf{D}} = [\text{vec}(\mathbf{D}_1), \text{vec}(\mathbf{D}_2), \dots, \text{vec}(\mathbf{D}_M)]$ . The real-valued manifold relation in (33) translates into the real-valued matrix relation

$$\mathbf{K}_{x1}\check{\mathbf{D}}\Omega_x = \mathbf{K}_{x2}\check{\mathbf{D}} \quad (40)$$

where  $\Omega_x = \text{diag}\{\tan(\frac{x_1}{2}), \dots, \tan(\frac{x_M}{2})\}$ . The Eq. (39) dictates that  $\check{\mathbf{D}}$  satisfies

$$\mathbf{K}_{y1}\check{\mathbf{D}}\Omega_y = \mathbf{K}_{y2}\check{\mathbf{D}} \quad (41)$$

where  $\Omega_y = \text{diag}\{\tan(\frac{y_1}{2}), \dots, \tan(\frac{y_M}{2})\}$ . Then we choose the  $M$  largest left singular vectors of  $\tilde{\mathbf{Y}}$  to form a  $L^2 \times M$  matrix  $\check{\mathbf{E}}$ . Approximately, we have

$$\check{\mathbf{E}} = \check{\mathbf{D}}\mathbf{T} \quad (42)$$

where  $\mathbf{T}$  is an unknown  $M \times M$  real-valued matrix. It is not difficult to see that substituting  $\check{\mathbf{D}} = \check{\mathbf{E}}\mathbf{T}^{-1}$  into (40) can get

$$\mathbf{K}_{x1}\check{\mathbf{E}}\Psi_x = \mathbf{K}_{x2}\check{\mathbf{E}} \quad (43)$$

where  $\Psi_x = \mathbf{T}^{-1}\Omega_x\mathbf{T}$ . And substituting  $\check{\mathbf{D}}$  into (41) yields the signal eigenvector relations

$$\mathbf{K}_{y1}\check{\mathbf{E}}\Psi_y = \mathbf{K}_{y2}\check{\mathbf{E}} \quad (44)$$

where  $\Psi_y = \mathbf{T}^{-1}\Omega_y\mathbf{T}$ . Thus, the eigenvalues of the  $M \times M$  matrix  $\Psi_x$  and  $\Psi_y$  correspond to solutions of the matrix equations in

(40) and (41), which are  $\tan(\frac{x_m}{2})$  and  $\tan(\frac{y_m}{2})$ ,  $m = 1, \dots, M$ , respectively. Since all of the quantities in (43) and (44) are real valued,  $\Psi = \Psi_x + j\Psi_y$  can be decomposed as

$$\Psi = \Psi_x + j\Psi_y = \mathbf{T}^{-1}(\Omega_x + j\Omega_y)\mathbf{T} \quad (45)$$

The eigenvalues  $\lambda_m$  of  $\Psi$  corresponds to  $\tan(\frac{x_m}{2}) + j\tan(\frac{y_m}{2})$ ,  $m = 1, \dots, M$ . Since the above processes provide closed-form, automatic pairing of final calculated  $x_m$  and  $y_m$ ,  $m = 1, \dots, M$  is facilitated. Hence, the positions of the new array elements can be accurately obtained, neither searching nor pairing is required.

After obtaining the new array, we can refer to  $\mathbf{Z}$  to construct a new matrix  $\mathbf{Z}_{new}$ . Next, the least square method is utilized to obtain the new excitation coefficient matrix  $\mathbf{W}_{new}$ , as follows:

$$\mathbf{W}_{new} = (\mathbf{Z}_{new}^T \mathbf{Z}_{new})^{-1} \mathbf{Z}_{new}^T \mathbf{Y} \quad (46)$$

According to Eq. (17), the desired multiple beampatterns are available.

### 3.3. Reducing the number of elements

In this subsection, we discuss how to determine the number of elements in the sparse new array. Usually, for an array of  $G$  elements, the SVD operation is performed on the data matrix  $\mathbf{Y}$  in (17), and we will obtain  $G$  non-zero singular values. However, observations have shown that for many designed antenna arrays in literature, the number of main singular values is less than that of antenna elements. This means that the contributions of some elements corresponding to non-primary singular values can be replaced by combinations of other elements. Therefore, we can discard the non-primary values to get a low-rank approximation matrix of  $\mathbf{Y}$ , which corresponds to a new antenna array with fewer elements. Hence, we perform SVD on matrix  $\mathbf{Y}$  as follows:

$$\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^H \quad (47)$$

where  $\mathbf{U} \in \mathbb{C}^{L^2 \times L^2}$  and  $\mathbf{V} \in \mathbb{C}^{P \times P}$  are unitary matrices. And

$$\Sigma = \begin{bmatrix} \Sigma_1 \\ \mathbf{0} \end{bmatrix} \in \mathbb{C}^{L^2 \times P} \quad (48)$$

where  $\Sigma_1 = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_S)$ , and  $\sigma_s$  is the ordered  $s$ th singular values of  $\mathbf{Y}$ , and  $S = \min\{P, L^2\}$ . Generally speaking, in order to contain the information of the initial beampatterns completely, the sampling points are not too few, so we might as well set  $L^2 > P$  here. Thus, we have  $S = P$ .

A typical low-rank approximation method is to set these non-primary singular values equal to zero, as follows:

$$\tilde{\mathbf{Y}} = \mathbf{U}\tilde{\Sigma}\mathbf{V}^H \quad (49)$$

where

$$\tilde{\Sigma} = \begin{bmatrix} \tilde{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{C}^{L^2 \times P} \quad (50)$$

where  $\tilde{\Sigma}_1 = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_M)$ . It has been proved that in all matrices of rank  $M$ ,  $\tilde{\mathbf{Y}}$  has the smallest approximation error in the Frobenius norm, that is, mathematically

$$\|\mathbf{Y} - \tilde{\mathbf{Y}}\|_F = \min_{\text{rank}(\mathbf{X})=M} \|\mathbf{Y} - \mathbf{X}\|_F = \sqrt{\sum_{s=M+1}^S \sigma_s^2} \quad (51)$$

From (51), we can know that the approximation error decreases monotonically as the number of new antenna elements  $M$  increases. When  $M = G$ , the approximation error approaches zero. This means that the radiation pattern of the new array can always achieve a good approximation.

In the practical problem, we use the following method to determine a suitable value of  $M$ :

$$M = \min\{M_{min}; |\frac{\sqrt{\sum_{s=M_{min}+1}^S \sigma_s^2}}{\sqrt{\sum_{s=1}^{M_{min}} \sigma_s^2}}| < \varepsilon\} \quad (52)$$

where  $\varepsilon$  is a small positive number. The determination of  $M$  is based on the error of reconstructed beampatterns approximating original radiation beampatterns. Then, the value of  $M$  obtained by Eq. (52) is substituted into the column number  $M$  of the  $\tilde{\mathbf{D}}$  matrix in (40). And the reconstructed planar array algorithm with automatic coordinate pairing is summarized in Algorithm 1.

---

#### Algorithm 1 The proposed Algorithm.

- 1: **Input:** the matrix  $\mathbf{Z}$  in (12) and  $\mathbf{W}$ , where  $\mathbf{Z}$  is the transpose of array manifold of the initial array, and  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_P]$  is composed of weight vectors of the  $P$  desired beampatterns
  - 2: calculate the sampling matrix of desired beampatterns  $\mathbf{Y}$  by (17)
  - 3: compute SVD of  $\mathbf{Y}$  by (47)
  - 4: determine the number of new array elements  $M$  in (52) by making a low-rank approximation to  $\mathbf{Y}$
  - 5: obtain  $\tilde{\mathbf{Y}}$  in (18) by  $\mathbf{Y}$  and unitarymatrix  $\mathbf{Q}_L$  defined in (5) and (6)
  - 6: compute SVD of  $\tilde{\mathbf{Y}}$  by (19)
  - 7: obtain  $\tilde{\mathbf{E}}$  composed of the  $M$  “largest” left singular vectors of  $\tilde{\mathbf{Y}}$
  - 8: compute  $\mathbf{K}_{x1} = \mathbf{I}_L \otimes \mathbf{K}_1$ ,  $\mathbf{K}_{x2} = \mathbf{I}_L \otimes \mathbf{K}_2$ ,  $\mathbf{K}_{y1} = \mathbf{K}_3 \otimes \mathbf{I}_L$  and  $\mathbf{K}_{y2} = \mathbf{K}_4 \otimes \mathbf{I}_L$ , where  $\mathbf{K}_1$  and  $\mathbf{K}_3$  are defined the same in (29),  $\mathbf{K}_2$  and  $\mathbf{K}_4$  are defined the same in (30) and  $\mathbf{I}_L$  is the identity matrix
  - 9: calculate  $\Psi_x$  and  $\Psi_y$  by (43) and (44)
  - 10: calculate the eigenvalues  $\lambda_m, m = 1, \dots, M$  of the matrix  $\Psi_x + j\Psi_y$
  - 11: compute the abscissa and ordinate of the automatic pairing of the new array elements,  $x_m = 2\tan^{-1}(\text{Re}(\lambda_m))$ ,  $y_m = 2\tan^{-1}(\text{Im}(\lambda_m))$ ,  $m = 1, \dots, M$
  - 12: calculate the new array manifold  $\mathbf{Z}_{new}$  of the new array
  - 13: obtain the new weight matrix  $\mathbf{W}_{new}$  of the new arraywith respect to  $P$  desired beampatterns from (46)
  - 14: **Output:** the coordinates  $(x_m, y_m), m = 1, \dots, M$  of the new array elements, the new weight matrix  $\mathbf{W}_{new}$  and the reconstructed beampatterns
- 

**Remark 1.** We assumed the rank of  $\tilde{\mathbf{Y}}$  is equal to  $G$  and the ranks of  $\Psi_x$  and  $\Psi_y$  are equal to  $M$  in Section 3. To ensure that these assumptions hold, the following conditions need to be met

$$L^2 \geq 2P = \tilde{S} > P = S \geq G \geq M \quad (53)$$

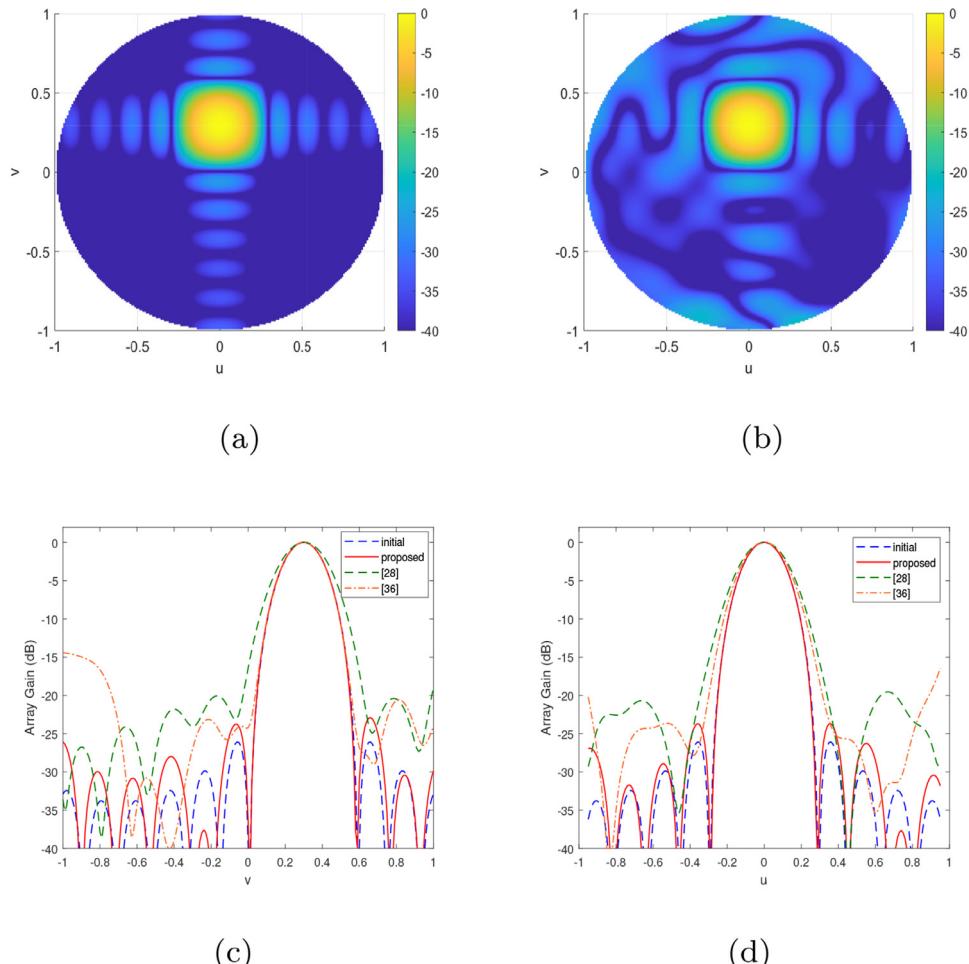
where  $L = 2N + 1$ .

### 3.4. Analysis

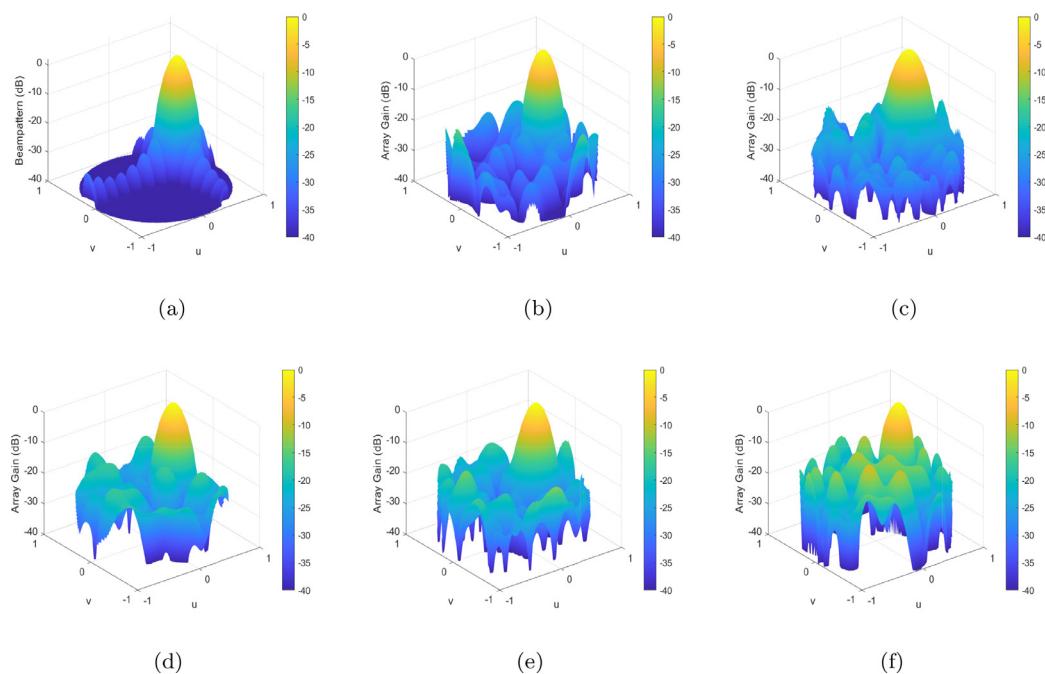
In this part, we analyze the time complexity and space complexity of the proposed algorithm.

#### 3.4.1. Time complexity

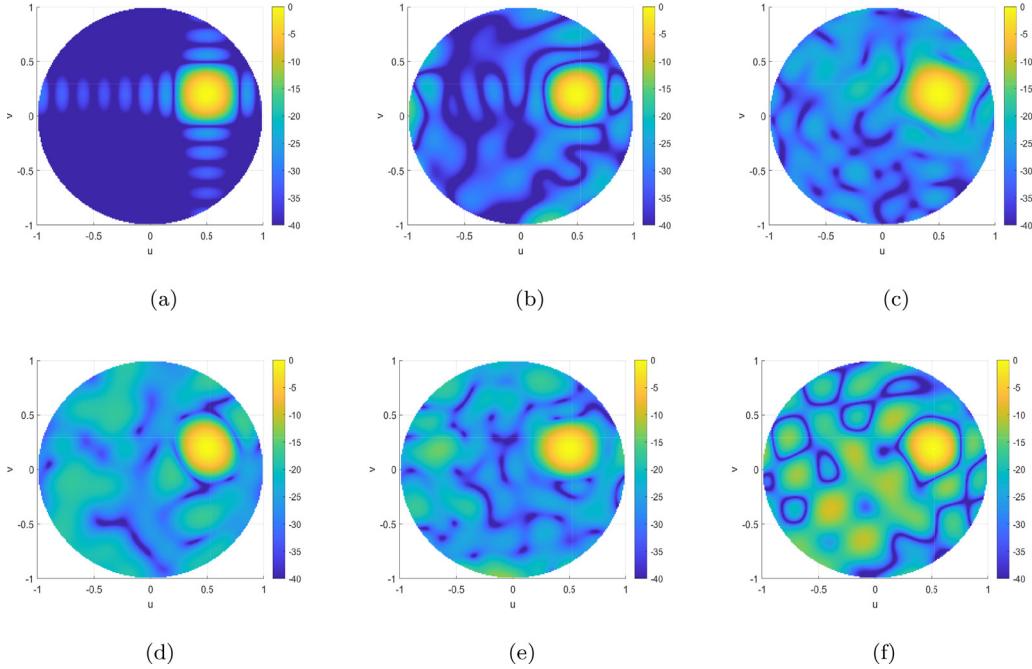
The SVD and matrix inversion processes occupy most of the computing resources in the implementation of the proposed method. From the Section 3 and (53), it can be known that the SVD of  $\tilde{\mathbf{Y}}$  and  $\mathbf{Y}$  is the most computationally complex, so the computational complexity of the proposed method is  $O(PL^4)$ . Likewise, the SVD process of GMEPM also occupies most of the computing resources. And the dimension of the decomposed matrix is  $G^2 \times PG^2$ , so the computational complexity is  $O(PG^6)$ , even after using randomized SVD, the computational complexity is  $O(PG^4 \log(\sqrt{G}))$ .  $L$



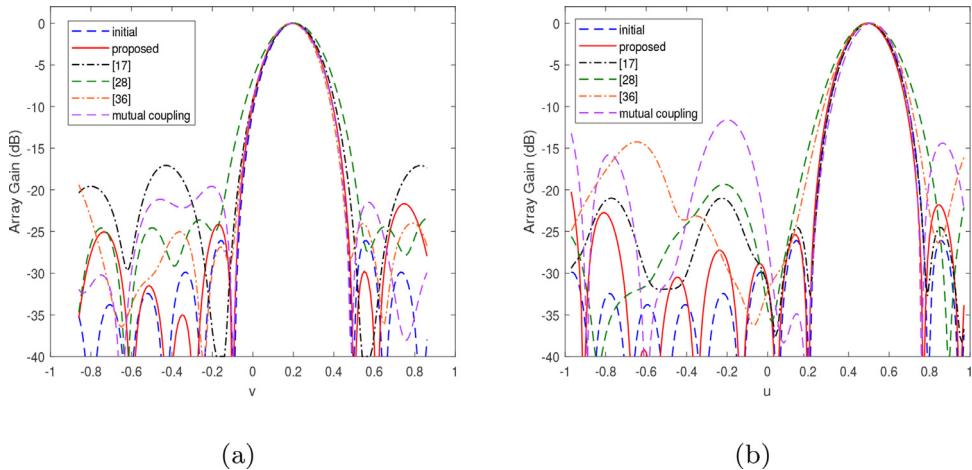
**Fig. 2.** Comparison of initial and reconstructed patterns at  $u=0$ ,  $v=0.3$ . (a) Top view of initial patterns. (b) Top view of reconstructed patterns. (c) Slice diagram at  $u=0$ . (d) Slice diagram at  $v=0.3$ .



**Fig. 3.** Comparison of 3-D view of initial and reconstructed patterns at  $u=0.5$ ,  $v=0.2$ . (a) 3-D view of initial pattern. (b) 3-D view of reconstructed pattern synthesized by the proposed method. (c) 3-D view of reconstructed pattern synthesized by the method in [28]. (d) 3-D view of reconstructed pattern synthesized by the method in [17]. (e) 3-D view of reconstructed pattern synthesized by the method in [20]. (f) 3-D view of reconstructed pattern synthesized by the proposed method considering the coupling.



**Fig. 4.** Comparison of top view of initial and reconstructed patterns at  $u=0.5$ ,  $v=0.2$ . (a) Top view of initial pattern. (b) Top view of reconstructed pattern synthesized by the proposed method. (c) Top view of reconstructed pattern synthesized by the method in [28]. (d) Top view of reconstructed pattern synthesized by the method in [17]. (e) Top view of reconstructed pattern synthesized by the method in [20]. (f) Top view of reconstructed pattern synthesized by the proposed method considering the coupling.



**Fig. 5.** Comparison of slice diagram of initial and reconstructed patterns at  $u=0.5$ ,  $v=0.2$ . (a) Slice diagram at  $u=0.5$ . (b) Slice diagram at  $v=0.2$ .

and  $G$  are of the same magnitude. Consequently, the computational complexity is lower than the method in [28].

#### 3.4.2. Space complexity

It is not difficult to see that the space occupied by matrix  $\tilde{\mathbf{Y}}$  is the largest in the proposed algorithm, so the space complexity of the proposed method is  $O(PL^2)$ . Similarly, in GMEPM of [28], a composite Hankel block matrix of size  $G^2 \times PG^2$  occupies largest space, so the space complexity of the proposed method is  $O(PG^4)$ . Hence, the space complexity of the proposed method also has an advantage over that of GMEPM.

Especially when the array is large, the advantage of the proposed method in complexity will be more obvious, since the complexity of [28] will increase rapidly with the increase of the number of array elements.

## 4. Numerical results

To more clearly evaluate the performance of the proposed algorithm, we set two parameters. They are the reduction rate  $\alpha$  of the number of array elements and the average error  $\epsilon$  of the reconstruction pattern, respectively. And the expressions are as follows:

$$\alpha = \frac{G - M}{G} \times 100\% \quad (54)$$

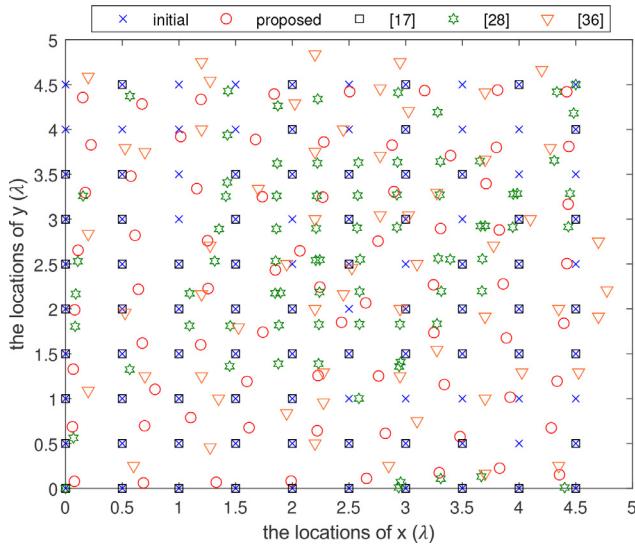
$$\epsilon = \frac{\sum_{n_1=-N}^N \sum_{n_2=-N}^N |\hat{\mathbf{F}}_{dB}(u_{n_1}, v_{n_2}) - \mathbf{F}_{dB}(u_{n_1}, v_{n_2})|}{(2N + 1)^2} \quad (55)$$

where  $\hat{\mathbf{F}}_{dB}(u_{n_1}, v_{n_2})$  and  $\mathbf{F}_{dB}(u_{n_1}, v_{n_2})$  represent the array gain (dB) at  $(u_{n_1}, v_{n_2})$  of the beampattern synthesized by the new array and

**Table 1**

Performance comparison of the proposed method and other methods with the focused beampatterns .

Parameters	Methods		
	proposed method	[28]	[17]
Initial 3dB beamwidth(u)	0.23	0.23	0.23
Reconstructed 3dB beamwidth(u)	0.23	0.27	0.25
Initial 3dB beamwidth(v)	0.23	0.23	0.23
Reconstructed 3dB beamwidth(v)	0.23	0.27	0.23
Target SLL	-20dB	-20dB	-20dB
Reconstructed average SLL	-23.28-dB	-21.79-dB	-18.59dB
Error $\epsilon$	1.71dB	9.67dB	13.26dB
Directivity (dBi)	22.11	20.58	21.93
Number of reconstructed array elements	71	76	71
Reduction ratio $\alpha$	29%	24%	29%
runtime	3.86s	511.08s	0.26s
Number of tested beampatterns	100	5	1

**Fig. 6.** Comparison of the positions of the new and initial elements for the focused patterns.

the initial array. Moreover, the directivity and mutual coupling of the planar array are also discussed. When the main beam of the planar array is steered to the direction of  $(\theta_0, \varphi_0)$ , the correspond-

ing directivity can be written as a matrix form, given by

$$D(\theta_0, \varphi_0) = \frac{\mathbf{w}_0^H \mathbf{a}_0 \mathbf{a}_0^H \mathbf{w}_0}{\mathbf{w}_0^H \mathbf{B} \mathbf{w}_0} \quad (56)$$

where  $\mathbf{w}_0$  and  $\mathbf{a}_0$  are the weight vector and steering vector corresponding to direction  $(\theta_0, \varphi_0)$ , respectively. And  $\mathbf{B}$  is a positive definite Hermite matrix determined by the array structure, which can be calculated by

$$B_{i,j} = \frac{\sin(\beta \rho_{ij})}{2\beta \rho_{ij}} \quad (57)$$

where

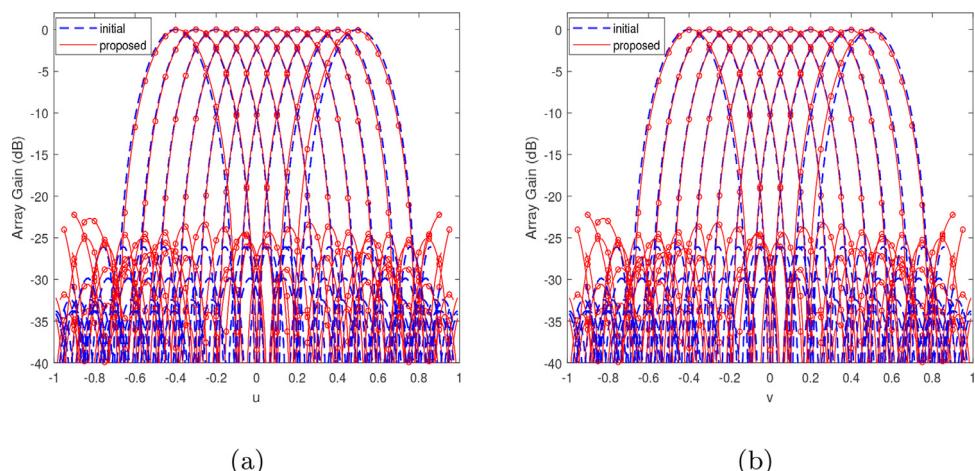
$$\rho_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (58)$$

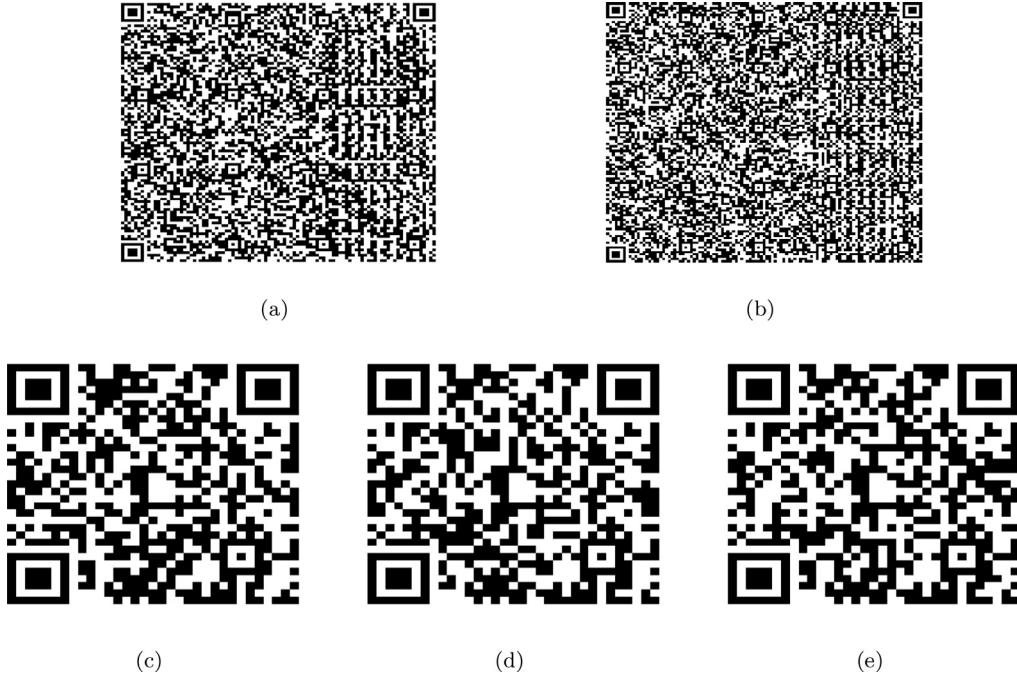
The specific expressions of the directivity can be obtained in [36]. And we use the nearest neighbor coupling coefficient to discuss the mutual coupling between radiators:

$$C_{(i,j)} = \begin{cases} \cos(\beta \Delta d_{(i,j)}), & \Delta d_{(i,j)} \leq \lambda/2 \\ 0, & \text{otherwise} \end{cases} \quad (59)$$

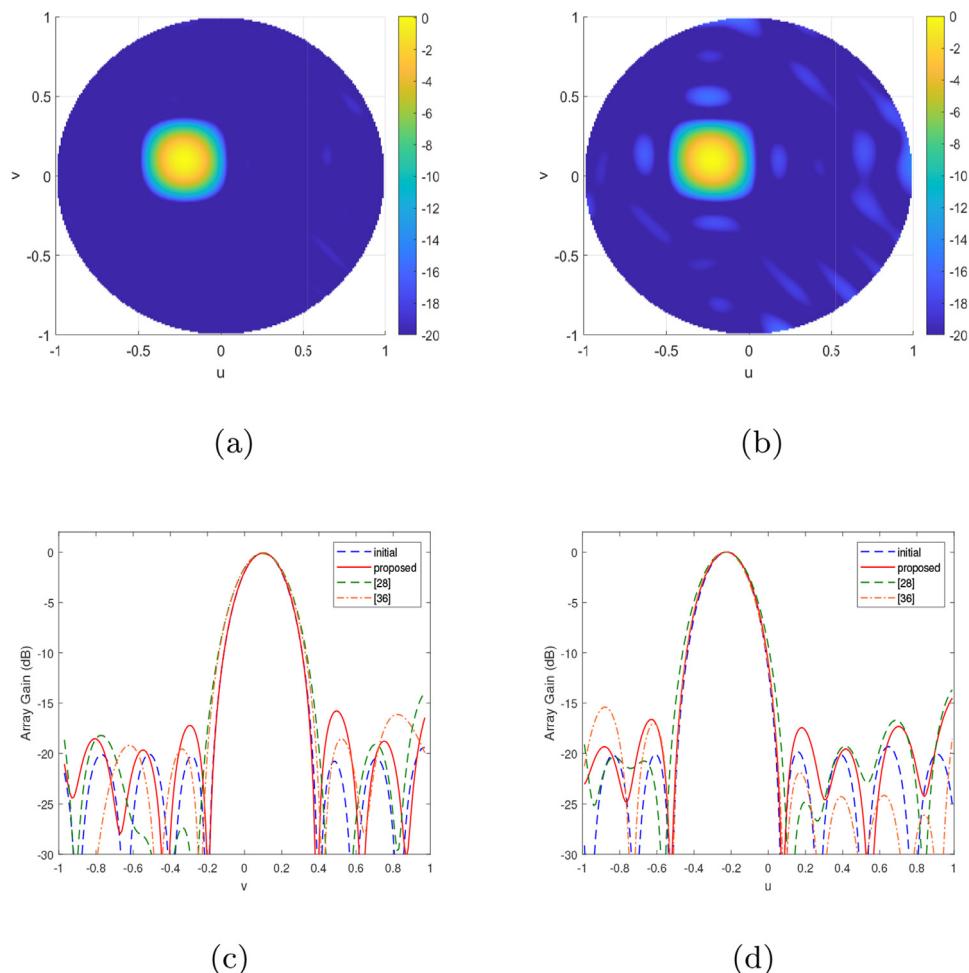
where  $\Delta d_{(i,j)}$  is the distance between  $i$ th and  $j$ th array element.

In this subsection, we present two simulation scenarios to evaluate the proposed algorithm. In the first scenario, all beampatterns are focused beampatterns with narrow mainlobe width and pointing at a certain angle. In the second scenario, half of the desired beampatterns are focused beampatterns, while the other half are flat-top beampatterns with a wide beamwidth. For both scenarios, the sidelobe levels (SLL) of all desired patterns are constrained to a specific level. To provide a more comprehensive evaluation of the

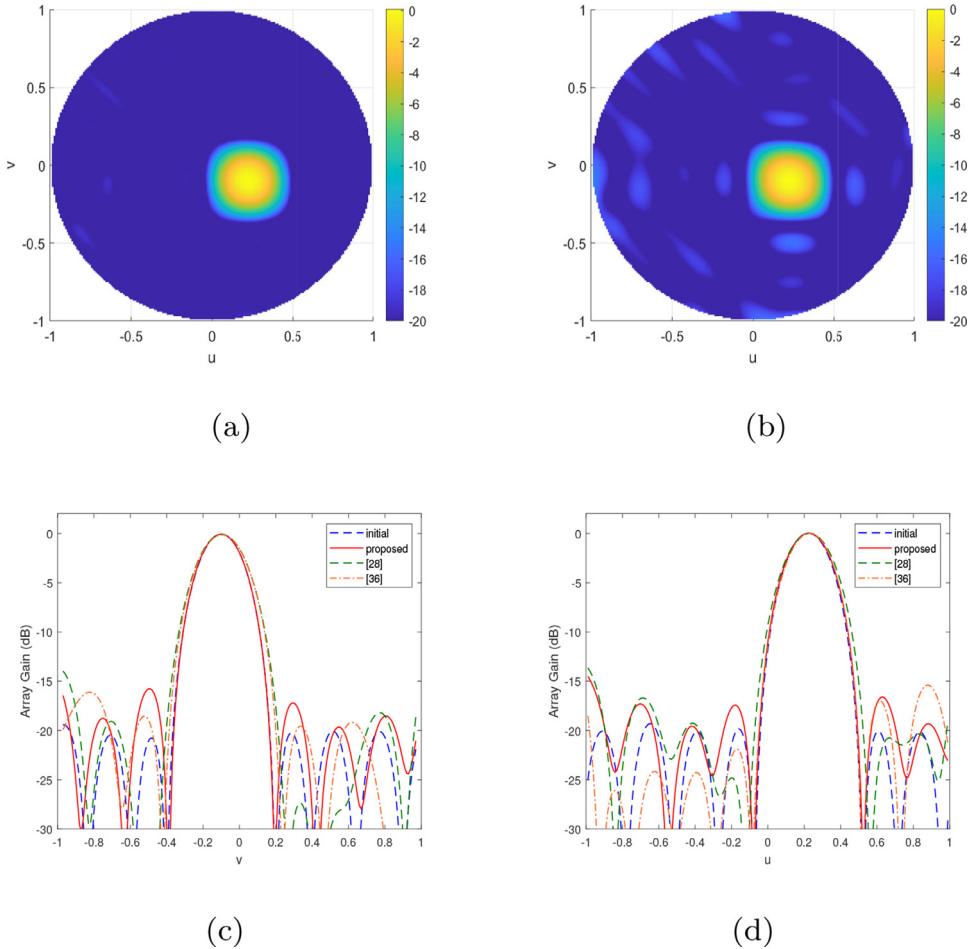
**Fig. 7.** Comparison of initial and reconstructed patterns. (a) Slice diagram at u. (b) Slice diagram at v.



**Fig. 8.** QR codes for the Taylor pattern. (a) the locations of  $x$ . (b) the locations of  $y$ . (c) excitations of Fig. 1. (d) excitations of Fig. 2. (e) excitations of Fig. 3.



**Fig. 9.** Comparison of initial and reconstructed patterns at  $u=-0.2$ ,  $v=0.1$ . (a) Initial patterns. (b) Reconstructed patterns. (c) Slice diagram at  $u=-0.2$ . (d) Slice diagram at  $v=0.1$ .



**Fig. 10.** Comparison of initial and reconstructed patterns at  $u=0.2$ ,  $v=-0.1$ . (a) Initial patterns. (b) Reconstructed patterns. (c) Slice diagram at  $u=0.2$ . (d) Slice diagram at  $v=-0.1$ .

proposed method, we compare it with three representative methods from the literature, namely the iterative FFT algorithm presented in [17], GMEPM presented in [28] and IDEA presented in [20].

#### 4.1. Focused beampatterns

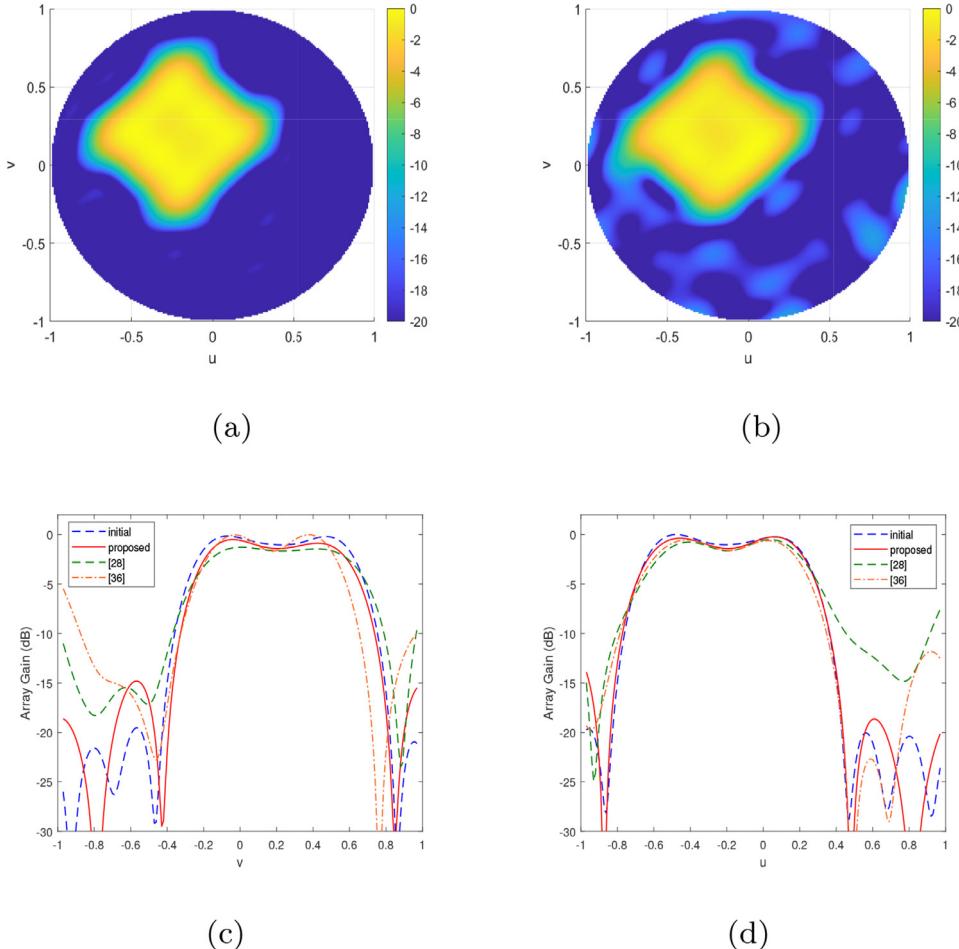
In this subsection, we consider an  $10 \times 10$  URA, that is  $Gr = 10$ ,  $Gc = 10$ , with  $\Delta_x = \Delta_y = \lambda/2$ . We selected  $P = 100$  beampatterns synthesized by Taylor window, whose mainlobe directions change uniformly in  $[u, v | u \in [-0.4, 0.5], v \in [-0.4, 0.5]]$ , and the sidelobe levels are all constrained to below -20 dB.

To demonstrate the performance of the proposed algorithm more comprehensively and thoroughly, we choose three beampatterns for comparison with three representative methods in [17,28] and in [20], which are far apart in the mainlobe directions. The three beam axes  $(u, v)$  are  $(-0.4, -0.3)$ ,  $(0, 0.3)$  and  $(0.5, 0.2)$ , as shown in Figs. 1-5. It should be emphasized that the proposed method is aimed at the synthesis of sparse planar arrays with multiple patterns, particularly when the number of beampatterns is large. However, there is no existing method with exactly the same function as this proposed method, so three algorithms with similar functions are selected. Although [28] also targets multi-beampattern, it requires too high time and space complexity to carry 100 beampatterns. Therefore, we tested it on only five beampatterns with mainlobe directions  $(u, v)$  of  $(-0.4, -0.4)$ ,  $(-0.4, -0.3)$ ,  $(0, 0.3)$ ,  $(0.5, 0.2)$  and  $(0.5, 0.5)$ . And IDEA is optimized according to the maximum scanning range of

the beam, so here we set the maximum scanning range  $u_{\max}$  as 0.5. On the other hand, since [17] can only operate on a single beampattern, only the result related to Fig. 3 is illustrated.

Figs. 1(a) and 1(b) show the initial and reconstructed two-dimensional top view of the corresponding beampattern, respectively. It can be observed that the reconstructed array can synthesize the desired pattern, albeit with increased SLL. This is due to the synthesis of desired patterns that simultaneously satisfy a wide range of different mainlobe directions. A slice comparison of the initial and reconstructed patterns in the direction of the mainlobe is shown in Fig. 1(c) and Fig. 1(d). The mainlobe before and after reconstruction by the proposed method are almost identical, with only minor differences in the sidelobes that do not significantly impact the mainlobe. Conversely, by GMEPM and IDEA, the error between the reconstructed beampattern and the initial beampattern is significant, especially in the mainlobe, which is noticeably widened. Fig. 2 compares the initial and reconstructed pattern of the mainlobe at  $(0, 0.3)$ . As shown in Fig. 2, the proposed algorithm can still synthesize the desired beampattern more accurately than the algorithm in [28] and [20].

In Fig. 3-Fig. 5, the simulation results of the algorithm in [17] are added and compared with GMEPM, IDEA and the proposed algorithm. To better demonstrate the performance of the four algorithms, we show the three-dimensional views of the initial beampattern and the beampattern synthesized by the four methods are respectively shown in Fig. 3(a)-Fig. 3(d). It is evident that the beampatterns synthesized by the four methods roughly fit the desired beampattern. However, beampatterns synthesized by the al-



**Fig. 11.** Comparison of initial and reconstructed patterns at  $u=-0.2$ ,  $v=0.2$ . (a) Initial patterns. (b) Reconstructed patterns. (c) Slice diagram at  $u=-0.2$ . (d) Slice diagram at  $v=0.2$ .

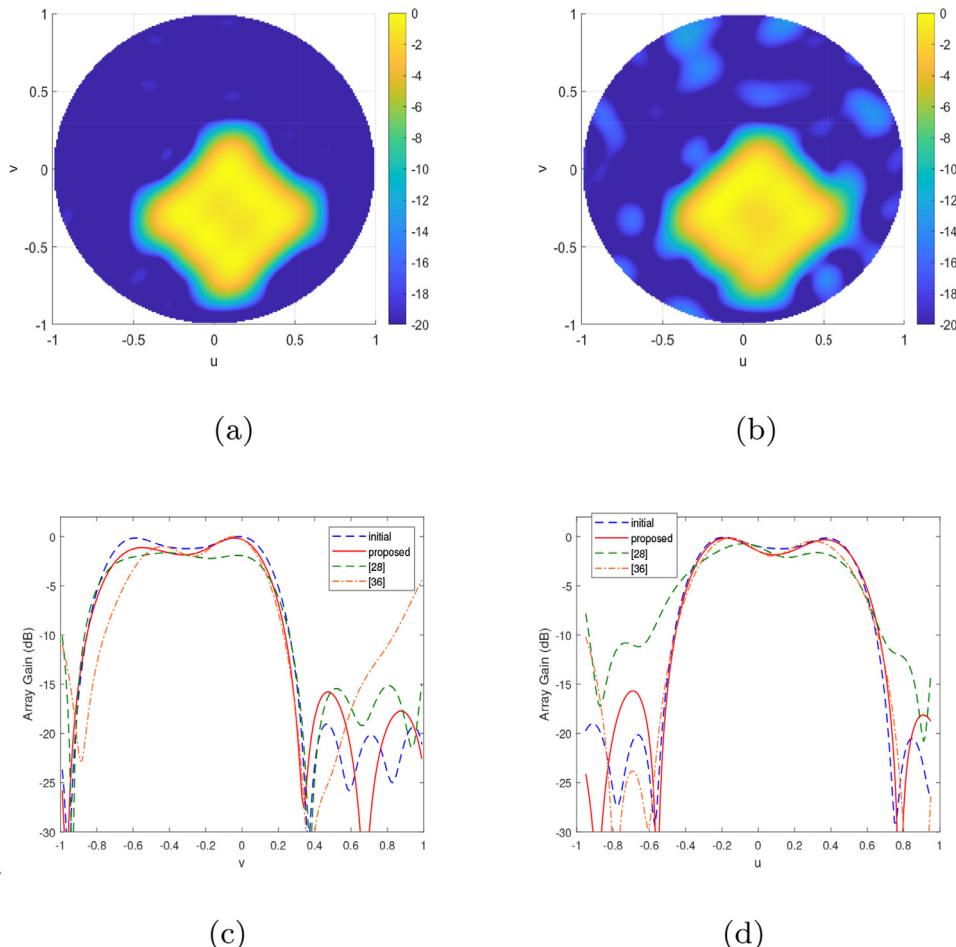
gorithm in [17,28] and [20] have more sidelobes with higher levels than the beampattern synthesized by the proposed algorithm. The top views of 3(a)-Fig. 3(d) are shown in 4(a)-Fig. 4(d). As we can see, the proposed algorithm has the best performance. The slice diagrams shown in 5(a) and Fig. 5(b) reveal that the mainlobes of beampatterns reconstructed by the proposed method and the method in [17] match the desired beampattern better than that of the method in [28] and [20]. However, the algorithm in [17] does not perform well on sidelobes. Moreover, the 3D view, top view and slice diagrams of the beampattern synthesized by the proposed method with mutual coupling are shown in Fig. 3(f), 4(f), 5(a) and 5(b).

Fig. 6 illustrates the positions of initial and reconstructed array elements. We observe that the aperture of the array reconstructed by the proposed method is slightly smaller than the original array. And it can be seen from the Fig. 6 that the array elements of the array reconstructed by the algorithm in [28] are unevenly distributed, resulting in a significant difference between the final beampatterns and the original ones. However, the array positions obtained by the algorithm in [17] are obviously still limited to the original array element positions, thus hindering the reconstruction of sidelobes. And the aperture of the array reconstructed by IDEA in [20] is bigger than the initial array. To further demonstrate the effectiveness of the proposed algorithm, we present additional comparisons of beampatterns in Fig. 7(a) and Fig. 7(b), pointing to angles such as  $(-0.4, -0.4)$ ,  $(-0.3, -0.3)$ ,  $(-0.2, -0.2)$ ,  $(-0.1, -0.1)$ ,  $(0, 0)$ ,  $(0.1, 0.1)$ ,  $(0.2, 0.2)$ ,  $(0.3, 0.3)$ ,  $(0.4, 0.4)$  and  $(0.5, 0.5)$ .

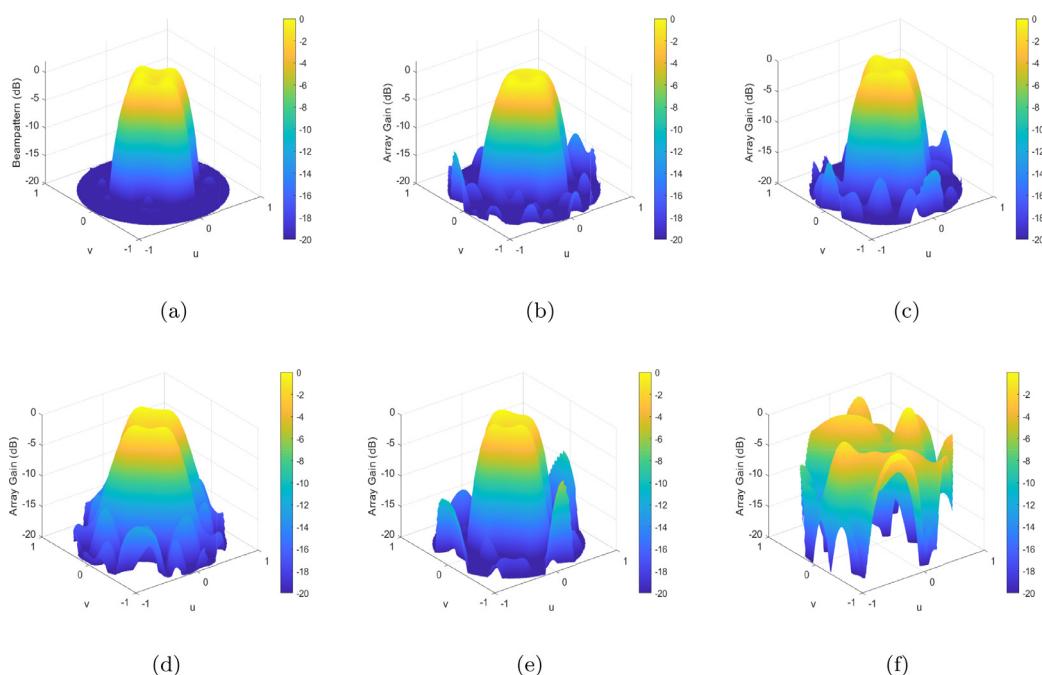
Table 1 provides detailed data, showing that the proposed method reduces the number of array elements from 100 to 71, achieving a reduction ratio of 29%. The average sidelobe level of the reconstructed beampattern of the proposed method, the method in [17,28] and [20] are  $-23.28\text{dB}$ ,  $-21.79\text{dB}$ ,  $-18.59\text{dB}$  and  $-21.03\text{dB}$ , respectively. The average SLL of the reconstructed beampattern of the proposed method is the best in the four methods. And the average error of the proposed method is also the smallest, only  $1.71\text{ dB}$ , while the other two methods have errors of  $9.67\text{ dB}$  and  $0.1326\text{ dB}$ . And the directivity obtained by the proposed method is  $22.11\text{dBi}$ , which is the highest. Although the time of the proposed method required to obtain the locations of the reconstructed array is  $3.86\text{s}$ , which is slightly larger than the time  $0.26\text{s}$  of the method in [17], it is much faster than the time  $511.08\text{s}$  and  $37\text{min}$  of the methods in [28] and [20]. Moreover, the proposed method ensures high reconstruction accuracy for multiple beampatterns simultaneously. It should be noted that the 3dB beamwidth, average SLL and error in Table 1 are only representative of the results in Fig. 3. Moreover, due to space constraints, the specific values of the array element positions synthesized by the proposed algorithm and the corresponding weight vectors are included in the QR codes in Fig. 8.

#### 4.2. Hybrid beampatterns

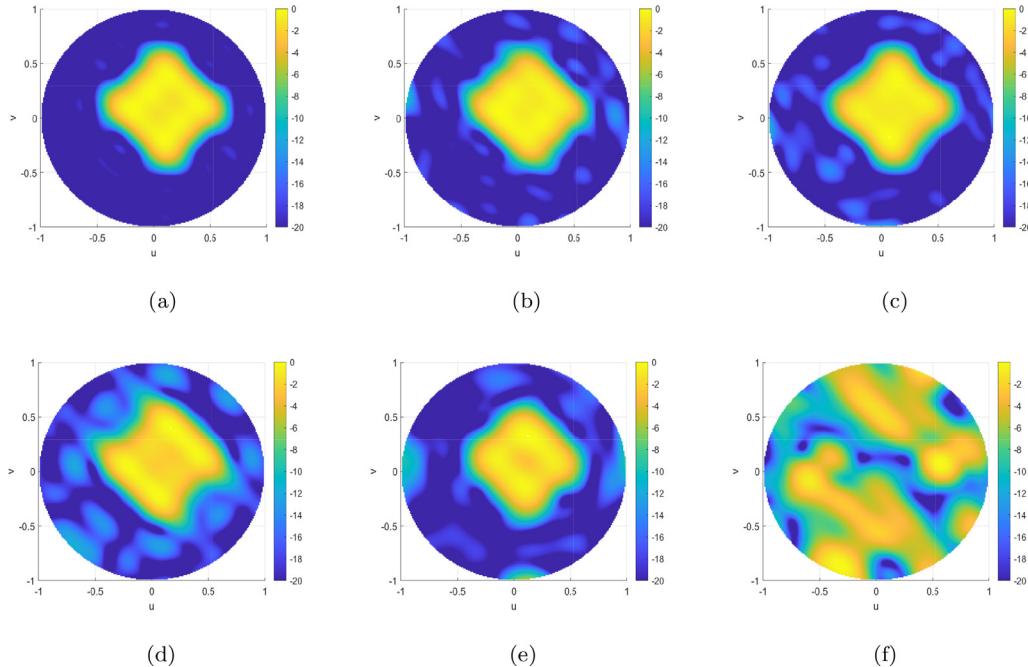
In this subsection, we consider an  $8 \times 8$  URA, that is  $Gr = 8$ ,  $Gc = 8$ , with  $\Delta_x = \Delta_y = \lambda/2$ . The initial desired beampatterns are generated using the WORD algorithm presented in [35].



**Fig. 12.** Comparison of initial and reconstructed patterns at  $u=0.1$ ,  $v=-0.3$ . (a) Initial patterns. (b) Reconstructed patterns. (c) Slice diagram at  $u=0.1$ . (d) Slice diagram at  $v=-0.3$ .



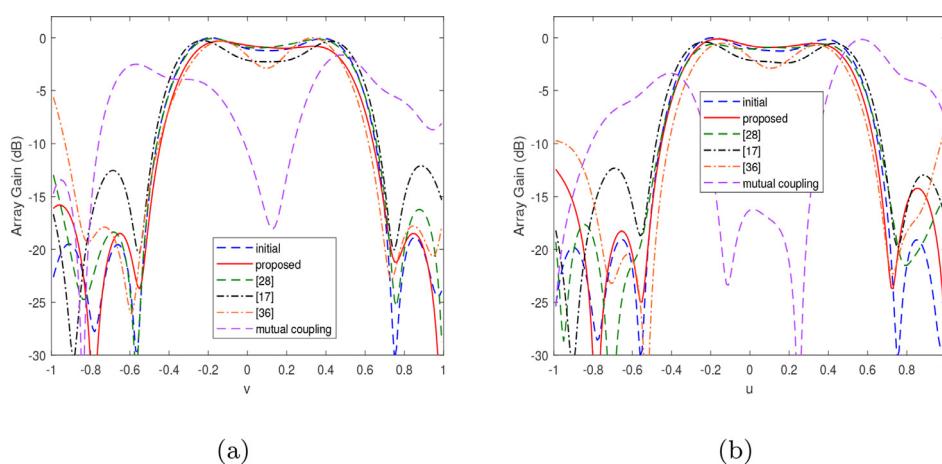
**Fig. 13.** Comparison of 3-D view of initial and reconstructed patterns at  $u=0.1$ ,  $v=0.1$ . (a) 3-D view of initial pattern. (b) 3-D view of reconstructed pattern synthesized by the proposed method. (c) 3-D view of reconstructed pattern synthesized by the method in [28]. (d) 3-D view of reconstructed pattern synthesized by the method in [17]. (e) 3-D view of reconstructed pattern synthesized by the method in [20]. (f) 3-D view of reconstructed pattern synthesized by the proposed method considering the coupling.



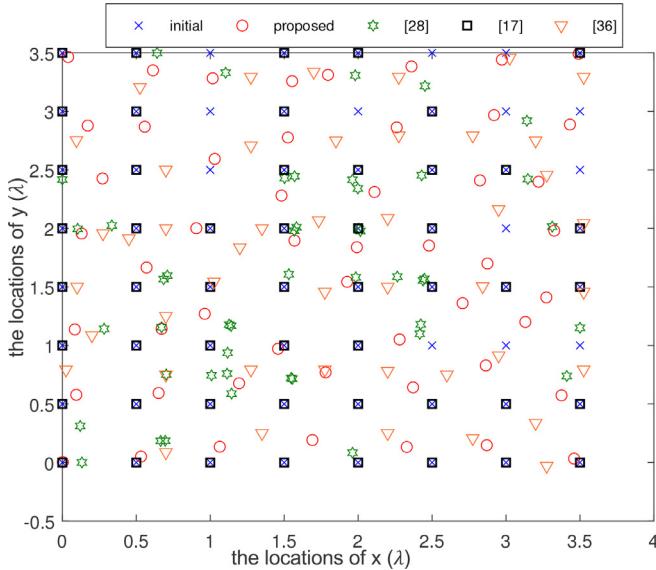
**Fig. 14.** Comparison of top view of initial and reconstructed patterns at  $u=0.1, v=0.1$ . (a) Top view of initial pattern. (b) Top view of reconstructed pattern synthesized by the proposed method. (c) Top view of reconstructed pattern synthesized by the method in [28]. (d) Top view of reconstructed pattern synthesized by the method in [17]. (e) Top view of reconstructed pattern synthesized by the method in [20]. (f) Top view of reconstructed pattern synthesized by the proposed method considering the coupling.

**Table 2**  
Performance comparison of the proposed method and other methods with the hybrid beampatterns .

Parameters	Methods			
	proposed method	[28]	[17]	[20]
Initial 3dB beamwidth( $u$ )	0.90	0.90	0.90	—
Reconstructed 3dB beamwidth( $u$ )	0.86	0.90	0.95	0.81
Initial 3dB beamwidth( $v$ )	0.91	0.91	0.91	—
Reconstructed 3dB beamwidth( $v$ )	0.85	0.91	0.95	0.83
Target SLL	-20dB	-20dB	-20dB	-20dB
Reconstructed average SLL	-16.67dB	-16.43dB	-14.65dB	-16.21dB
Error $\epsilon$	0.69dB	0.78dB	2.42dB	—
Directivity (dBi)	12.96	12.87	10.94	11.06
Number of reconstructed array elements	51	46	51	45
Reduction ratio $\alpha$	20.3%	28.1%	20.3%	—
runtime	8.56s	375.38s	0.42s	12min
Number of tested beampatterns	70	5	1	—



**Fig. 15.** Comparison of slice diagram of initial and reconstructed patterns at  $u=0.1, v=0.1$ . (a) Slice diagram at  $u=0.1$ . (b) Slice diagram at  $v=0.1$ .



**Fig. 16.** Comparison of the positions of the new and initial elements for the hybrid patterns.

We selected  $P = 70$  beampatterns, of which the mainlobe directions of 35 beampatterns vary uniformly in  $[u, v] | u \in [-0.2, 0.2], v \in [-0.3, 0.3]$ . And the remaining 35 beampatterns are flat-top beampatterns, the central axes of which also change uniformly in  $[u, v] | u \in [-0.2, 0.2], v \in [-0.3, 0.3]$ , and the flat-top area is defined as a square with a diagonal of 0.4. Moreover, the sidelobe levels of all beampatterns are constrained to below -20 dB.

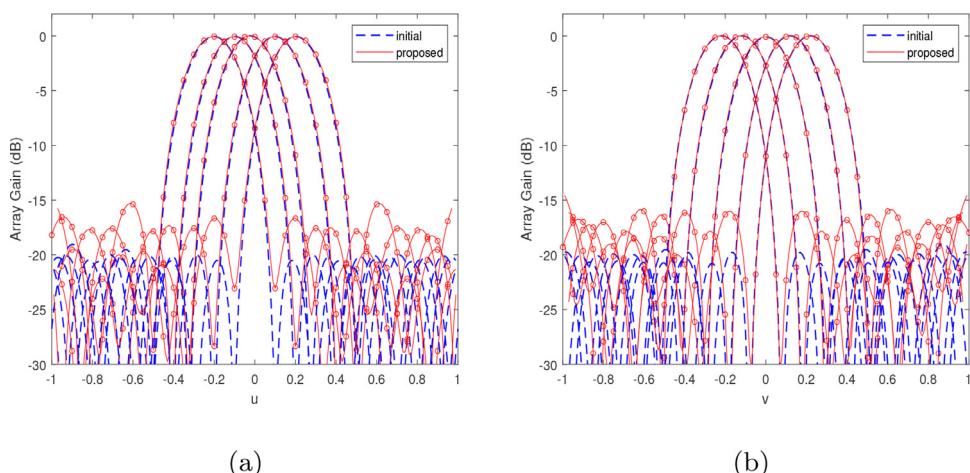
Two focused and three flat-top beampatterns are selected for presentation in this example, and their beam axes ( $u, v$ ) are  $(-0.2, 0.1)$ ,  $(0.2, -0.1)$ ,  $(-0.2, 0.2)$ ,  $(0.1, -0.3)$  and  $(0.1, 0.1)$ , respectively. These two reconstruction focused patterns are presented in Fig. 9 and Fig. 10. They again confirm the effectiveness of the proposed algorithm for focused beampatterns, even in the hybrid multi-beampattern. The three flat-top beampatterns are shown in the Fig. 11–Fig. 15. Similarly, the method in [28] cannot carry 70 beampatterns, so only five beampatterns demonstrated in Fig. 9–Fig. 15 were selected for testing. Since [17] can only process single beampattern, only the result related to Fig. 13 is displayed.

From Fig. 11(a) and Fig. 11(b), it can be observed that the reconstructed flat-top pattern is generally consistent with the initial beampattern. Moreover, the slices in Fig. 11(c) and 11(d) clearly

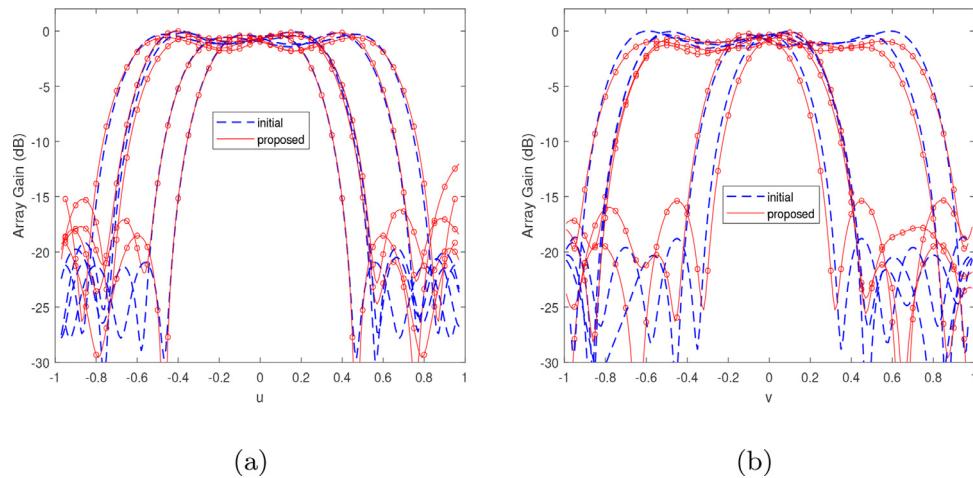
show that the flat-top area of the reconstructed beampattern and that of the initial beampattern basically overlap, despite the SLL increases. On the other hand, the beampatterns reconstructed by the algorithm in [28] and [20] differ significantly from the initial beampattern, particularly in the flat-top area. Fig. 12 presented comparison of initial and reconstructed flat-top pattern of the mainlobe at  $(0.1, -0.3)$ . And performance comparison results displayed in Fig. 12 are similar to those in Fig. 11.

In Fig. 13–Fig. 15, the results obtained by the method in [17] are added to compare with the algorithm in [20,28] and the proposed algorithm. Fig. 13(a)–Fig. 13(e) show the three-dimensional views of the initial beampattern and the beampatterns synthesized by the four methods. And the corresponding top views are shown in Fig. 14(a)–Fig. 14(e). It is evident that the beampatterns synthesized by the three algorithms closely resemble the initial beampattern. However, the beampattern synthesized by the algorithm in [17] has more sidelobes with higher level values than the beampattern synthesized by the proposed algorithm. From the slice diagrams shown in 15(a) and Fig. 15(b), there are significant differences between the beampatterns reconstructed by the algorithm in [28] and the initial pattern in the flat-top region, and so is IDEA. These results confirm the effectiveness of the proposed algorithm in synthesizing flat-top beampatterns.

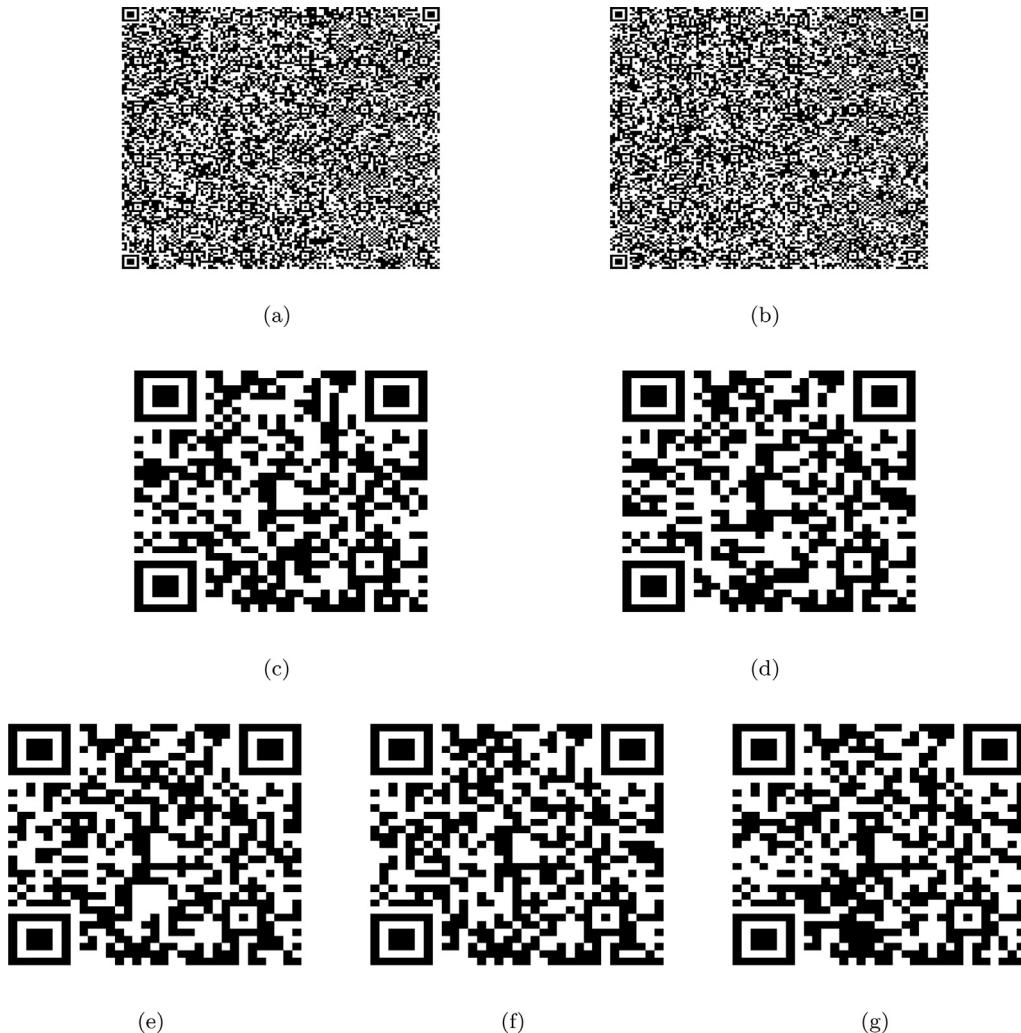
The elements distribution of the reconstructed and initial array is displayed in Fig. 16. The aperture of the array reconstructed by the proposed method is also slightly smaller than the original array. Compared with the proposed method, the array obtained by the algorithm in [28] has a more concentrated array elements distribution, leading to a larger difference between the reconstructed and the original beampatterns. Meanwhile, the algorithm in [17] does not perform well in the sidelobes due to the limited positions of its array elements on the original grids, and the optimal performance cannot be obtained. Moreover, in Fig. 17(a) and Fig. 17(b), there are more comparisons of the focused beampatterns demonstrated, and their pointing angles are  $(-0.2, -0.2)$ ,  $(-0.1, -0.1)$ ,  $(0, 0)$ ,  $(0.1, 0.1)$ , and  $(0.2, 0.2)$ , respectively. Similarly, more flat-top beampatterns are shown in Fig. 18(a) and Fig. 18(b), and their central angles are  $(-0.2, 0.3)$ ,  $(-0.1, -0.2)$ ,  $(-0.1, 0.2)$ ,  $(0.1, -0.2)$  and  $(0.2, -0.3)$ . These results demonstrate the effectiveness of the proposed algorithm in both focused and flat-top beampatterns, compared to the existing methods. Moreover, the 3D view, top view and slice diagrams of the beampattern synthesized by the proposed method with mutual coupling are shown in Fig. 13(f), 14(f), 15(a) and 15(b).



**Fig. 17.** Comparison of initial and reconstructed patterns. (a) Slice diagram at  $u$ . (b) Slice diagram at  $v$ .



**Fig. 18.** Comparison of initial and reconstructed patterns. (a) Slice diagram at  $u$ . (b) Slice diagram at  $v$ .



**Fig. 19.** QR codes for the WORD pattern. (a) the locations of  $x$ . (b) the locations of  $y$ . (c) excitations of Fig. 9. (d) excitations of Fig. 10. (e) excitations of Fig. 11. (f) excitations of Fig. 12. (g) excitations of Fig. 13.

In Table 2, we present additional data to further demonstrate the performance of the proposed method. The number of array elements is reduced from 70 to 51, resulting in a reduction ratio of 20.3%. The average SLL of the reconstructed beampattern using the proposed method is -16.67dB, which is better than that

of -16.43dB, -14.65dB and -16.21dB obtained by the algorithms in [17,28] and [20]. Moreover, the average error of the proposed method, the algorithm in [28] and [17] are 0.69dB, 0.78dB and 2.42dB, respectively, with the proposed method having the smallest average error. And the directivity obtained by the proposed

method is 12.96dBi, which is the highest. The time required by the proposed method to obtain the locations of the reconstructed array is 8.56s, while the time required by the method in [17,28] and [20] are 375.38s, 0.42s and 12min, respectively. It should be noted that the 3dB beamwidth, average SLL and error in Table 2 are only representative of the results in Fig. 13. Moreover, the specific values of the array element positions synthesized by the proposed algorithm and the corresponding weight vectors are included in the QR codes in Fig. 19.

## 5. Conclusion

In this paper, we have derived the relationship between the positions of the array elements and the column space of the data matrix based on unitary ESPRIT. Meanwhile, the low rank approximation has been used to reduce the number of reconstructed array elements. Finally, we have proposed a two-dimensional closed-form method for the synthesis of sparse planar arrays with multi-beampattern. The proposed method achieves automatic coordinate pairing of the array elements between X-axis and Y-axis. And the proposed algorithm saves 20% to 30% of elements, while maintaining an acceptable approximation error. Compared with the existing algorithm, the algorithm can handle a larger number of desired patterns and does not require additional pairing searches. This results in improved synthesis accuracy, as well as cost and time savings due to its low complexity. Future work can investigate the application of the proposed method in different scenarios and explore the potential of combining it with other beamforming techniques.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## CRediT authorship contribution statement

**Yangjingzhi Zhuang:** Conceptualization, Formal analysis, Validation, Writing – original draft, Writing – review & editing. **Xuejing Zhang:** Supervision, Writing – review & editing. **Zishu He:** Supervision, Writing – review & editing.

## Data availability

No data was used for the research described in the article.

## Acknowledgements

This work was supported in part by the National Nature Science Foundation of China under Grants 62101101, 62031007 and 62231006, and in part by the Peng Cheng Shang Xue Education Fund of UESTC Education Development Foundation under Grant XY2021602.

## References

- [1] A. Chakraborty, B.N. Das, G.S. Sanyal, Beam shaping using nonlinear phase distribution in a uniformly spaced array, *IEEE Trans. Antennas Propag.* AP-30 (5) (1982) 1031–1034.
- [2] O.M. Bucci, G. Mazzarella, G. Panariello, Reconfigurable arrays by phase-only control, *IEEE Trans. Antennas Propag.* 39 (7) (1991) 919–925.
- [3] C.A. Balanis, *Antenna Theory: Analysis and Design*, 3rd, Wiley, New York, NY, USA, 2005.
- [4] Z. Jiang, S. Xiao, Y. Li, A wide-angle time-domain electronically scanned array based on energy-pattern-reconfigurable elements, *IEEE Antennas Wireless Propag. Lett.* 17 (9) (2018) 1598–1602.
- [5] J. Row, C. Tsai, Pattern reconfigurable antenna array with circular polarization, *IEEE Trans. Antennas Propag.* 64 (4) (2016) 1525–1530.
- [6] B.P. Kumar, G.R. Branner, Generalized analytical technique for the synthesis of unequally spaced arrays with linear, planar, cylindrical or spherical geometry, *IEEE Trans. Antennas Propag.* 53 (2) (2005) 621–634.
- [7] E. Boudaher, F. Ahmad, M.G. Amin, Mutual coupling effect and compensation in non-uniform arrays for direction-of-arrival estimation, *Digit. Signal Process.* 61 (2017) 3–14.
- [8] J. Costantine, Y. Tawk, S.E. Barbin, C.G. Christodoulou, Reconfigurable antennas: design and applications, *Proc. IEEE.* 103 (3) (2015) 424–437.
- [9] R.L. Haupt, Thinned arrays using genetic algorithms, *IEEE Trans. Antennas Propag.* 42 (7) (1994) 993–999.
- [10] V. Murino, A. Trucco, C.S. Regazzoni, Synthesis of equally spaced arrays by simulated annealing, *IEEE Trans. Signal Process.* 44 (1) (1996) 119–122.
- [11] D.W. Boeringer, D.H. Werner, Particle swarm optimization versus genetic algorithms for phased array synthesis, *IEEE Trans. Antennas Propag.* 52 (3) (2004) 771–779.
- [12] Y. Bai, S. Xiao, C. Liu, B. Wang, A hybrid IWO/PSO algorithm for pattern synthesis of conformal phased arrays, *IEEE Trans. Antennas Propag.* 61 (4) (2013) 2328–2332.
- [13] C. Jiang, F. Hu, F. He, J. Li, D. Zhu, Low-redundancy large linear arrays synthesis for aperture synthesis radiometers using particle swarm optimization, *IEEE Trans. Antennas Propag.* 64 (6) (2016) 2179–2188.
- [14] S. Yang, Y.B. Gan, A. Qing, Sideband suppression in time-modulated linear arrays by the differential evolution algorithm, *IEEE Antennas Wireless Propag. Lett.* 1 (1) (2002) 173–175.
- [15] S. Caorsi, A. Massa, M. Pastorino, A. Randazzo, Optimization of the difference patterns for monopulse antennas by a hybrid real/integer-coded differential evolution method, *IEEE Trans. Antennas Propag.* 53 (1) (2005) 372–376.
- [16] F. Zhang, W. Jia, M. Yao, Linear aperiodic array synthesis using differential evolution algorithm, *IEEE Antennas Wireless Propag. Lett.* 12 (2013) 797–800.
- [17] W.P.M.N. Keizer, Linear array thinning using iterative FFT techniques, *IEEE Trans. Signal Process.* 56 (8) (2008) 2757–2760.
- [18] W.P.M.N. Keizer, Synthesis of thinned planar circular and square arrays using density tapering, *IEEE Trans. Antennas Propag.* 62 (4) (2014) 1555–1563.
- [19] S.E. Nai, W. Ser, Z.L. Yu, H. Chen, Beam pattern synthesis for linear and planar arrays with antenna selection by convex optimization, *IEEE Trans. Antennas Propag.* 58 (12) (2010) 3923–3930.
- [20] D. Pinchera, M.D. Migliore, G. Panariello, Synthesis of large sparse arrays using IDEA (inflating-deflating exploration algorithm), *IEEE Trans. Antennas Propag.* 66 (9) (2018) 4658–4668.
- [21] B. Fuchs, Synthesis of sparse arrays with focused or shaped beampattern via sequential convex optimizations, *IEEE Trans. Antennas Propag.* 60 (7) (2012) 3499–3503.
- [22] Y. Hua, T.K. Sarkar, Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise, *IEEE Trans. Antennas Propag.* 38 (5) (1990) 814–824.
- [23] Y. Liu, Z. Nie, Q.H. Liu, Reducing the number of elements in a linear antenna array by the matrix pencil method, *IEEE Trans. Antennas Propag.* 56 (9) (2008) 2955–2962.
- [24] Y. Liu, Q.H. Liu, Z. Nie, Reducing the number of elements in the synthesis of shaped-beam patterns by the forward-backward matrix pencil method, *IEEE Trans. Antennas Propag.* 58 (2) (2010) 604–608.
- [25] Y. Liu, Q.H. Liu, Z. Nie, Reducing the number of elements in multiple-pattern linear arrays by the extended matrix pencil methods, *IEEE Trans. Antennas Propag.* 62 (2) (2014) 652–660.
- [26] Y. Hua, Estimating two-dimensional frequencies by matrix enhancement and matrix pencil, *IEEE Trans. Signal Process.* 40 (9) (1992) 2267–2280.
- [27] P. Gu, G. Wang, Z. Fan, R. Chen, An efficient approach for the synthesis of large sparse planar array, *IEEE Trans. Antennas Propag.* 67 (12) (2019) 7320–7330.
- [28] Y. Gong, S. Xiao, B. Wang, Synthesis of sparse planar arrays with multiple patterns by the generalized matrix enhancement and matrix pencil, *IEEE Trans. Antennas Propag.* 69 (2) (2021) 869–881.
- [29] G. Oliveri, A. Massa, Bayesian compressive sampling for pattern synthesis with maximally sparse non-uniform linear arrays, *IEEE Trans. Antennas Propag.* 59 (2) (2011) 467–481.
- [30] G. Oliveri, M. Carlin, A. Massa, Complex-weight sparse linear array synthesis by bayesian compressive sampling, *IEEE Trans. Antennas Propag.* 60 (5) (2012) 2309–2326.
- [31] F. Viani, G. Oliveri, A. Massa, Compressive sensing pattern matching techniques for synthesizing planar sparse arrays, *IEEE Trans. Antennas Propag.* 61 (9) (2013) 4577–4587.
- [32] G. Oliveri, E.T. Bekelle, F. Robol, A. Massa, Sparsening conformal arrays through a versatile BCS-based method, *IEEE Trans. Antennas Propag.* 62 (4) (2014) 1681–1689.
- [33] M. Carlin, G. Oliveri, A. Massa, Hybrid BCS-deterministic approach for sparse concentric ring isophoric arrays, *IEEE Trans. Antennas Propag.* 63 (1) (2015) 378–383.
- [34] M.D. Zoltowski, M. Haardt, C.P. Mathews, Closed-form 2-d angle estimation with rectangular arrays in element space or beamspace via unitary esprit, *IEEE Trans. Signal Process.* 44 (2) (1996) 316–328.
- [35] X. Zhang, Z. He, B. Liao, et al., Pattern synthesis for arbitrary arrays via weight vector orthogonal decomposition, *IEEE Trans. Signal Process.* 66 (5) (2018) 1286–1299.
- [36] F. Yang, Y. Ma, W. Long, et al., Synthesis of irregular phased arrays subject to constraint on directivity via convex optimization, *IEEE Trans. Antennas Propag.* 69 (7) (2021) 4235–4240.