Hindawi
International Journal of Antennas and Propagation
Volume 2017, Article ID 6862852, 7 pages
https://doi.org/10.1155/2017/6862852

Research Article

Joint-2D-SL0 Algorithm for Joint Sparse Matrix Reconstruction

Dong Zhang, Yongshun Zhang, and Cunqian Feng

Air and Missile Defense College, Air Force Engineering University, Xi'an, Shaanxi 710051, China

Correspondence should be addressed to Dong Zhang; zhangdongtougao@163.com

Received 19 July 2017; Revised 4 November 2017; Accepted 26 November 2017; Published 20 December 2017

Academic Editor: Elisa Giusti

Sparse matrix reconstruction has a wide application such as DOA estimation and STAP. However, its performance is usually restricted by the grid mismatch problem. In this paper, we revise the sparse matrix reconstruction model and propose the joint sparse matrix reconstruction model based on one-order Taylor expansion. And it can overcome the grid mismatch problem. Then, we put forward the Joint-2D-SL0 algorithm which can solve the joint sparse matrix reconstruction problem efficiently. Compared with the Kronecker compressive sensing method, our proposed method has a higher computational efficiency and acceptable reconstruction accuracy. Finally, simulation results validate the superiority of the proposed method.

1. Introduction

Compressive sensing is becoming more and more popular for its superiority in parameter super-resolution estimation using short observation [1–3]. And as extensions to compressed sensing, sparse matrix reconstruction has received a lot of attention [4–6]. Many problems in signal processing can be seemed as sparse matrix reconstruction problem, such as the DOA estimation [7] and STAP [8]. In this paper, we consider the estimation of DOA and DOD in MIMO radar. It can be solved by the traditional subspace method, such as MUSIC and ESPRIT algorithm. But they usually need large snapshots to estimate the covariance matrix. Considering the advantages of sparse matrix reconstruction, here, we research the estimation of DOA and DOD in MIMO radar based on sparse matrix reconstruction method. And many algorithms have been proposed to solve the sparse matrix reconstruction efficiently. For instance, [9] puts forward the 2D-SL0 algorithm, and [10] puts forward the 2D-IAA algorithm. Both of them can reconstruct the sparse matrix efficiently. However, the sparse matrix model has some inherent shortcomings. Its performance is affected by the grid mismatch problem [11]. That is because no matter how thin we divide the mesh, we still cannot guarantee that all the parameters fall on the grid completely [12, 13]. So the estimation accuracy will be affected by the grid number and how much do we divide the mesh.

In this paper, we revise the sparse matrix model by the one-order Taylor expansion and propose the joint sparse matrix model. This model eliminates the grid mismatch effect by introducing some joint sparse items. Then, in order to solve the joint sparse matrix reconstruction problem efficiently, we revise the 2D-SL0 algorithm and put forward the Joint-2D-SL0 algorithm. It can get a high estimation accuracy with satisfied speed.

Note that our method is different with the methods in [9, 10]. The methods in [9, 10] cannot deal with the off-grid problem. Our method can solve it by introducing some joint sparse items. And these sparse items have the same sparse structure. So the "Joint" means that our Joint-2D-SL0 algorithm can get their estimation simultaneously. However, the "Joint" in [14] means that it is jointly used to reconstruct the images at all available channels simultaneously. That is the difference between our method and the method in [14]. Both our method and the method in [14] are applied to the signal matrix without stacking the signal into 1D vector.

2. Problem Formulation

Assuming there is a bistatic MIMO radar which obtains K transmitters and L receivers, both the transmitting array and the receiving array are collocated in uniform linear array. d_1 and d_2 are the element spacing of transmitters and receivers, respectively. $f^{(\mathrm{T})} = d_1\cos(\varphi)/\lambda$ and $f^{(\mathrm{R})} = d_2\cos(\theta)/\lambda$ are the normalized DOD and normalized DOA of targets. For convenience, we denote them as the DOD and DOA in our paper. Assuming the transmitting waveforms are normalized orthogonal signals, that is, $SS^H = I_K$, where $S = [s_1, s_2, \ldots, s_K]^T$. After matched filtering, we get the received signal as follows:

$$\begin{aligned}\mathbf{Y} &= \left[\sum_{m=1}^{M} a_m \mathbf{a}_{\mathrm{R}}\left(f_m^{(\mathrm{R})}\right)\mathbf{a}_{\mathrm{T}}\left(f_m^{(\mathrm{T})}\right)^{\mathrm{T}} S + \mathbf{W}\right] S^H \\ &= \sum_{m=1}^{M} a_m \mathbf{a}_{\mathrm{R}}\left(f_m^{(\mathrm{R})}\right)\mathbf{a}_{\mathrm{T}}\left(f_m^{(\mathrm{T})}\right)^{\mathrm{T}} + \mathbf{Z},\end{aligned} \tag{1}$$

where a_m is the scattering coefficient of mth target. $\mathbf{W}$ represents the noise matrix. $\mathbf{Z} = \mathbf{W}S^H$. $\mathbf{a}_{\mathrm{R}}(f_m^{(\mathrm{R})}) = [1, \ldots, \exp(j(L-1)2\pi f_m^{(\mathrm{R})})]$ and $\mathbf{a}_{\mathrm{T}}(f_m^{(\mathrm{T})}) = [1, \ldots, \exp(j(\mathrm{K}-1)2\pi f_m^{(\mathrm{T})})]$ are the receiving steering vectors and transmitting steering vectors corresponding to the DOA and DOD of the mth target, respectively.

Considering the sparsity of targets, the sparse reconstruction method can be used to estimate target's 2D parameters. Then, discretizing the range of DOD and DOA to $K_d > K$ and $L_a > L$ resolution grids, respectively, we can convert (1) into the following form:

$$\mathbf{Y} = \mathbf{\Phi}_{\mathrm{R}}(\theta)\Xi\mathbf{\Phi}_{\mathrm{T}}{}^{\mathrm{T}}(\varphi) + \mathbf{Z}, \tag{2}$$

where Ξ is a spare matrix and we can estimate the DOD and DOA of targets according to the position of nonzero elements. $\theta = [\theta_1, \theta_2, \ldots, \theta_{K_d}]$ and $\varphi = [\varphi_1, \varphi_2, \ldots, \varphi_{L_a}]$ are the predefined grids. $\mathbf{\Phi}_{\mathrm{R}}(\theta) = [\mathbf{a}_{\mathrm{R}}(f_1^{(\mathrm{R})}), \mathbf{a}_{\mathrm{R}}(f_2^{(\mathrm{R})}), \ldots, \mathbf{a}_{\mathrm{R}}(f_{K_d}^{(\mathrm{R})})]$ and $\mathbf{\Phi}_{\mathrm{T}}(\varphi) = [\mathbf{a}_{\mathrm{T}}(f_1^{(\mathrm{T})}), \mathbf{a}_{\mathrm{T}}(f_2^{(\mathrm{T})}), \ldots, \mathbf{a}_{\mathrm{T}}(f_{L_a}^{(\mathrm{T})})]$.

This sparse matrix reconstruction model requires that the targets must fall on the predefined grids. Practically, no matter how small we divide the mesh, the targets cannot be guaranteed to completely fall on the grids. So the performance of this method will depend on the way how we divide the mesh. Reference [11] puts forward an off-grid model to solve the grid mismatch problem in 1D-DOA estimation; this idea could be extended to the 2D parameter estimation situation.

3. Joint Sparse Matrix Reconstruction

The targets' true and unknown DOA and DOD are $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_D]$ and $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_D]$, respectively. Approximating the $\mathbf{\Phi}_{\mathrm{R}}(\boldsymbol{\alpha})$ and $\mathbf{\Phi}_{\mathrm{T}}(\boldsymbol{\beta})$ by the first-order Taylor expansion around the predefined grids θ and φ, respectively,

$$\begin{aligned}\mathbf{\Phi}_{\mathrm{R}}(\boldsymbol{\alpha}) &= \mathbf{\Phi}_{\mathrm{R}}(\theta) + \mathbf{\Phi}_{\mathrm{R}}(\theta)'\mathbf{\Delta}_1, \\ \mathbf{\Phi}_{\mathrm{T}}(\boldsymbol{\beta}) &= \mathbf{\Phi}_{\mathrm{T}}(\varphi) + \mathbf{\Phi}_{\mathrm{T}}(\varphi)'\mathbf{\Delta}_2,\end{aligned} \tag{3}$$

where $\mathbf{\Phi}_{\mathrm{R}}(\theta)' = [(\partial\mathbf{a}_{\mathrm{R}}(\theta_1)/\partial\theta_1), \ldots, (\partial\mathbf{a}_{\mathrm{R}}(\theta_{L_a})/\partial\theta_{L_a})]$, $\mathbf{\Phi}_{\mathrm{T}}(\varphi)' = [(\partial\mathbf{a}_{\mathrm{T}}(\varphi_1)/\partial\varphi_1), \ldots, (\partial\mathbf{a}_{\mathrm{T}}(\varphi_{K_d})/\partial\varphi_{K_d})]$, $\mathbf{\Delta_1} = \mathrm{diag}(\boldsymbol{\alpha} - \theta)$, and $\mathbf{\Delta}_2 = \mathrm{diag}(\boldsymbol{\beta} - \varphi)$. We can get the following joint sparse matrix model:

$$\begin{aligned}\mathbf{Y} &= \left[\mathbf{\Phi}_{\mathrm{R}}(\theta) + \mathbf{\Phi}_{\mathrm{R}}(\theta)'\mathbf{\Delta}_1\right]\Xi\left[\mathbf{\Phi}_{\mathrm{T}}(\varphi) + \mathbf{\Phi}_{\mathrm{T}}(\varphi)'\mathbf{\Delta}_2\right]^{\mathrm{T}} + \mathbf{Z} \\ &= \left[\mathbf{\Phi}_{\mathrm{R}}(\theta) \quad \mathbf{\Phi}_{\mathrm{R}}(\theta)'\right]\begin{bmatrix} I \\ \mathbf{\Delta}_1 \end{bmatrix}\Xi\left[I \quad \mathbf{\Delta}_2{}^{\mathrm{T}}\right]\left[\mathbf{\Phi}_{\mathrm{T}}(\varphi) \quad \mathbf{\Phi}_{\mathrm{T}}(\varphi)'\right]^{\mathrm{T}} + \mathbf{Z} \\ &= \left[\mathbf{\Phi}_{\mathrm{R}}(\theta) \quad \mathbf{\Phi}_{\mathrm{R}}(\theta)'\right]\begin{bmatrix} \Xi & \Xi\mathbf{\Delta}_2{}^{\mathrm{T}} \\ \mathbf{\Delta}_1\Xi & \mathbf{\Delta}_1\Xi\mathbf{\Delta}_2{}^{\mathrm{T}} \end{bmatrix}\left[\mathbf{\Phi}_{\mathrm{T}}(\varphi) \quad \mathbf{\Phi}_{\mathrm{T}}(\varphi)'\right]^{\mathrm{T}} + \mathbf{Z} \\ &= \left[\mathbf{\Phi}_{\mathrm{R}}(\theta) \quad \mathbf{\Phi}_{\mathrm{R}}(\theta)'\right]\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix}\left[\mathbf{\Phi}_{\mathrm{T}}(\varphi) \quad \mathbf{\Phi}_{\mathrm{T}}(\varphi)'\right]^{\mathrm{T}} + \mathbf{Z},\end{aligned} \tag{4}$$

where $\mathbf{P}_1 = \mathbf{\Delta}_1\Xi$, $\mathbf{P}_2 = \Xi\mathbf{\Delta}_2{}^{\mathrm{T}}$, $\mathbf{P}_3 = \mathbf{\Delta}_1\Xi\mathbf{\Delta}_2{}^{\mathrm{T}}$, and Ξ are joint sparse matrix, that is, they have the same sparse structure. That is because whether left multiplied or right multiplied, a diagonal matrix will not change the sparsity of a sparse matrix. So we can use (4) to estimate the targets' DOD and DOA, and it will not be affected by the grid mismatch problem.

Using $\mathbf{\Phi}_{\mathrm{R}}$ and $\mathbf{\Phi}_{\mathrm{T}}$ to represent $\mathbf{\Phi}_{\mathrm{R}}(\theta)$ and $\mathbf{\Phi}_{\mathrm{T}}(\varphi)$, respectively, (4) can be rewritten as follows:

$$\mathbf{Y} = \left[\mathbf{\Phi}_{\mathrm{R}} \quad \mathbf{\Phi}_{\mathrm{R}}{}'\right]\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix}\left[\mathbf{\Phi}_{\mathrm{T}} \quad \mathbf{\Phi}_{\mathrm{T}}{}'\right]^{\mathrm{T}} + \mathbf{Z}. \tag{5}$$

But how to efficiently solve the joint sparse reconstruction problem? If we use the Kronecker compressive sensing method (solve (5) by converting it into 1D problem), it will bring much more computation burden because of the huge computation complexity of Kronecker product. Can we directly solve the joint sparse matrix reconstruction problem in (5)? Reference [9] proposes the 2D smoothed L0 (2D-SL0) algorithm, and it solves the 2D sparse problem more easily than the 1D-SL0 algorithm. Based on this, we revise the 2D-SL0 algorithm and propose the Joint-2D-SL0 algorithm to solve the joint sparse matrix reconstruction problem. They can be applied to situations where the number of targets is unknown.

The Gaussian function adopted in the 2D-SL0 is $G_\sigma(\mathbf{X}) = \sum_{i,j}\exp(-(x_{i,j}{}^2)/\sigma^2)$ [9]. Here, the Gaussian function we adopt in the Joint-2D-SL0 algorithm is

$$G_\sigma\left(\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix}\right) = \sum_{i,j}\exp\left(-\frac{\Xi_{i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2}{\sigma^2}\right). \tag{6}$$

It will lead to a joint sparse of $\Xi, \mathbf{P}_1, \mathbf{P}_2$, and $\mathbf{P}_3$, that is, the special case of block sparse of $\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix}$. The joint 2D gradient projection method is put forward to solve the Joint-2D-SL0 function. The gradient of the Joint-2D-SL0 function is

$$\delta \triangleq \nabla G_\sigma\left(\begin{bmatrix}\Xi & \mathbf{P}_2\\ \mathbf{P}_1 & \mathbf{P}_3\end{bmatrix}\right) = \begin{bmatrix} -\Xi_{i,j}\exp\left(-\dfrac{\Xi_{i,j}^{\ 2}+\mathbf{P}_{1i,j}^{\ 2}+\mathbf{P}_{2i,j}^{\ 2}+\mathbf{P}_{3i,j}^{\ 2}}{\sigma^2}\right) & \cdots & -\mathbf{P}_{2i,j}\exp\left(-\dfrac{\Xi_{i,j}^{\ 2}+\mathbf{P}_{1i,j}^{\ 2}+\mathbf{P}_{2i,j}^{\ 2}+\mathbf{P}_{3i,j}^{\ 2}}{\sigma^2}\right)\\ \vdots & \ddots & \vdots \\ -\mathbf{P}_{1i,j}\exp\left(-\dfrac{\Xi_{i,j}^{\ 2}+\mathbf{P}_{1i,j}^{\ 2}+\mathbf{P}_{2i,j}^{\ 2}+\mathbf{P}_{3i,j}^{\ 2}}{\sigma^2}\right) & \cdots & -\mathbf{P}_{3i,j}\exp\left(-\dfrac{\Xi_{i,j}^{\ 2}+\mathbf{P}_{1i,j}^{\ 2}+\mathbf{P}_{2i,j}^{\ 2}+\mathbf{P}_{3i,j}^{\ 2}}{\sigma^2}\right)\end{bmatrix}, \quad (7)$$

where σ_1 is chosen as $\sigma_1 > 8\max_{i,j}|\mathbf{W}_{i,j}|$ and $\mathbf{W} = \begin{bmatrix}\Xi^0 & \mathbf{P}_2^{\ 0}\\ \mathbf{P}_1^{\ 0} & \mathbf{P}_3^{\ 0}\end{bmatrix}$.

The initialization of the joint sparse matrix is

$$\begin{bmatrix}\Xi^0 & \mathbf{P}_2^{\ 0}\\ \mathbf{P}_1^{\ 0} & \mathbf{P}_3^{\ 0}\end{bmatrix} = \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}^\dagger \mathbf{Y} \begin{bmatrix}\mathbf{\Phi}_\mathrm{T}^{\ \mathrm{T}}\\ \mathbf{\Phi}_\mathrm{T}'^{\mathrm{T}}\end{bmatrix}^\dagger . \quad (8)$$

The proof is as follows:

Proof

$$\begin{aligned}\operatorname{vec}\left(\begin{bmatrix}\Xi^0 & \mathbf{P}_2^{\ 0}\\ \mathbf{P}_1^{\ 0} & \mathbf{P}_3^{\ 0}\end{bmatrix}\right) &= \left(\begin{bmatrix}\mathbf{\Phi}_\mathrm{T} & \mathbf{\Phi}_\mathrm{T}'\end{bmatrix} \otimes \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}\right)^\dagger \operatorname{vec}(\mathbf{Y})\\ &= \left(\begin{bmatrix}\mathbf{\Phi}_\mathrm{T} & \mathbf{\Phi}_\mathrm{T}'\end{bmatrix}^\dagger \otimes \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}^\dagger\right) \operatorname{vec}(\mathbf{Y})\\ &= \operatorname{vec}\left(\begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}^\dagger \mathbf{Y} \begin{bmatrix}\mathbf{\Phi}_\mathrm{T}^{\ \mathrm{T}}\\ \mathbf{\Phi}_\mathrm{T}'^{\mathrm{T}}\end{bmatrix}^\dagger\right).\end{aligned} \quad (9)$$

The projection onto the feasible set can be obtained by

$$\begin{aligned}\begin{bmatrix}\Xi & \mathbf{P}_2\\ \mathbf{P}_1 & \mathbf{P}_3\end{bmatrix} &= \begin{bmatrix}\Xi^l & \mathbf{P}_2^{\ l}\\ \mathbf{P}_1^{\ l} & \mathbf{P}_3^{\ l}\end{bmatrix} + \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}^\dagger\\ &\cdot\left\{\mathbf{Y} - \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}\begin{bmatrix}\Xi^l & \mathbf{P}_2^{\ l}\\ \mathbf{P}_1^{\ l} & \mathbf{P}_3^{\ l}\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_\mathrm{T}^{\ \mathrm{T}}\\ \mathbf{\Phi}_\mathrm{T}'^{\mathrm{T}}\end{bmatrix}\right\}\\ &\cdot\begin{bmatrix}\mathbf{\Phi}_\mathrm{T}^{\ \mathrm{T}}\\ \mathbf{\Phi}_\mathrm{T}'^{\mathrm{T}}\end{bmatrix}^\dagger .\end{aligned} \quad (10)$$

The proof is as follows:

Proof. When reconstructing a sparse matrix using the model $\mathbf{Y} = \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}\begin{bmatrix}\Xi^l & \mathbf{P}_2^{\ l}\\ \mathbf{P}_1^{\ l} & \mathbf{P}_3^{\ l}\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_\mathrm{T} & \mathbf{\Phi}_\mathrm{T}'\end{bmatrix}^\mathrm{T} + \mathbf{Z}$, we have

$$\begin{aligned}&\mathbf{Y} - \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}\begin{bmatrix}\Xi^l & \mathbf{P}_2^{\ l}\\ \mathbf{P}_1^{\ l} & \mathbf{P}_3^{\ l}\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_\mathrm{T} & \mathbf{\Phi}_\mathrm{T}'\end{bmatrix}^\mathrm{T}\\ &= \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}\begin{bmatrix}\Xi & \mathbf{P}_2\\ \mathbf{P}_1 & \mathbf{P}_3\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_\mathrm{T} & \mathbf{\Phi}_\mathrm{T}'\end{bmatrix}^\mathrm{T}\\ &\quad - \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}\begin{bmatrix}\Xi^l & \mathbf{P}_2^{\ l}\\ \mathbf{P}_1^{\ l} & \mathbf{P}_3^{\ l}\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_\mathrm{T} & \mathbf{\Phi}_\mathrm{T}'\end{bmatrix}^\mathrm{T}\\ &= \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}\left\{\begin{bmatrix}\Xi & \mathbf{P}_2\\ \mathbf{P}_1 & \mathbf{P}_3\end{bmatrix} - \begin{bmatrix}\Xi^l & \mathbf{P}_2^{\ l}\\ \mathbf{P}_1^{\ l} & \mathbf{P}_3^{\ l}\end{bmatrix}\right\}\begin{bmatrix}\mathbf{\Phi}_\mathrm{T} & \mathbf{\Phi}_\mathrm{T}'\end{bmatrix}^\mathrm{T}.\end{aligned} \quad (11)$$

So the minimum L_2 estimate of $\left\{\begin{bmatrix}\Xi & \mathbf{P}_2\\ \mathbf{P}_1 & \mathbf{P}_3\end{bmatrix} - \begin{bmatrix}\Xi^l & \mathbf{P}_2^{\ l}\\ \mathbf{P}_1^{\ l} & \mathbf{P}_3^{\ l}\end{bmatrix}\right\}$ is

$$\begin{aligned}&\begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}^\dagger\left\{\mathbf{Y} - \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}\begin{bmatrix}\Xi^l & \mathbf{P}_2^{\ l}\\ \mathbf{P}_1^{\ l} & \mathbf{P}_3^{\ l}\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_\mathrm{T} & \mathbf{\Phi}_\mathrm{T}'\end{bmatrix}^\mathrm{T}\right\}\\ &\quad\cdot\left\{\begin{bmatrix}\mathbf{\Phi}_\mathrm{T} & \mathbf{\Phi}_\mathrm{T}'\end{bmatrix}^\mathrm{T}\right\}^\dagger .\end{aligned} \quad (12)$$

So the projection onto the feasible set can be obtained by

$$\begin{aligned}\begin{bmatrix}\Xi & \mathbf{P}_2\\ \mathbf{P}_1 & \mathbf{P}_3\end{bmatrix} &= \begin{bmatrix}\Xi^l & \mathbf{P}_2^{\ l}\\ \mathbf{P}_1^{\ l} & \mathbf{P}_3^{\ l}\end{bmatrix} + \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}^\dagger\\ &\quad\cdot\left\{\mathbf{Y} - \begin{bmatrix}\mathbf{\Phi}_\mathrm{R} & \mathbf{\Phi}_\mathrm{R}'\end{bmatrix}\begin{bmatrix}\Xi^l & \mathbf{P}_2^{\ l}\\ \mathbf{P}_1^{\ l} & \mathbf{P}_3^{\ l}\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_\mathrm{T}^{\ \mathrm{T}}\\ \mathbf{\Phi}_\mathrm{T}'^{\mathrm{T}}\end{bmatrix}\right\}\\ &\quad\cdot\begin{bmatrix}\mathbf{\Phi}_\mathrm{T}^{\ \mathrm{T}}\\ \mathbf{\Phi}_\mathrm{T}'^{\mathrm{T}}\end{bmatrix}^\dagger .\end{aligned} \quad (13)$$

Now we get the Joint-2D-SL0 algorithm which is shown in Algorithm 1.

Remark 1. The internal loop is repeated a fixed and small number of times (L). That is to say, for increasing the speed, we do not wait for the steepest ascent algorithm to converge. This can be justified by the gradual decrease in the value of σ. And for each σ, we do not need the exact maximizer of G_σ. We just need to enter the region near the (global) maximizer of G_σ to escape from its local maximizers.

Initialize:

(1) Let $\begin{bmatrix} \Xi^0 & \mathbf{P}_2{}^0 \\ \mathbf{P}_1{}^0 & \mathbf{P}_3{}^0 \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_R & \mathbf{\Phi}_R{}' \end{bmatrix}^\dagger \boldsymbol{Y} \begin{bmatrix} \mathbf{\Phi}_T{}^T \\ \mathbf{\Phi}_T{}'^T \end{bmatrix}^\dagger$.

(2) Choose a suitable decreasing sequence for σ, $[\sigma_1, \sigma_2, \ldots, \sigma_J]$.

For $j = 1, 2, \ldots, J$:

(1) Let $\sigma = \sigma_J$

(2) Minimise the function $G_\sigma\left(\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix}\right) = \sum_{i,j} \exp(-\frac{\Xi_{i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2}{\sigma^2})$,

on the feasible set $\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} = \left\{ \begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} : \left| \begin{bmatrix} \mathbf{\Phi}_R & \mathbf{\Phi}_R{}' \end{bmatrix} \begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_T{}^T \\ \mathbf{\Phi}_T{}'^T \end{bmatrix} - \mathbf{Y} \right|_2 < \varepsilon \right\}$,

using L iterations of the steepest descent algorithm (then project $\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix}$ onto the feasible set):

Initialize: $\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} = \begin{bmatrix} \Xi^{j-1} & \mathbf{P}_2^{j-1} \\ \mathbf{P}_1^{j-1} & \mathbf{P}_3^{j-1} \end{bmatrix}$

For $l = 1, 2, \ldots, L$

(a) Let

$$\delta \triangleq \sigma^2 \nabla G_\sigma\left(\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix}\right) = \begin{bmatrix} -\Xi_{i,j}\exp(-\frac{\Xi_{i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2}{\sigma^2}) & \cdots & -\mathbf{P}_{2i,j}\exp(-\frac{\Xi_{i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2}{\sigma^2}) \\ \vdots & \ddots & \vdots \\ -\mathbf{P}_{1i,j}\exp(-\frac{\Xi_{i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2}{\sigma^2}) & \cdots & -\mathbf{P}_{3i,j}\exp(-\frac{\Xi_{i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2}{\sigma^2}) \end{bmatrix}.$$

(b) Let $\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} \leftarrow \begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} + \mu\delta$ (μ is a small positive constant)

(c) Project $\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix}$ back onto the feasible set:

$$\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} \leftarrow \begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} + \begin{bmatrix} \mathbf{\Phi}_R & \mathbf{\Phi}_R{}' \end{bmatrix}^\dagger \left\{ \mathbf{Y} - \begin{bmatrix} \mathbf{\Phi}_R & \mathbf{\Phi}_R{}' \end{bmatrix} \begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_T{}^T \\ \mathbf{\Phi}_T{}'^T \end{bmatrix} \right\} \begin{bmatrix} \mathbf{\Phi}_T{}^T \\ \mathbf{\Phi}_T{}'^T \end{bmatrix}^\dagger.$$

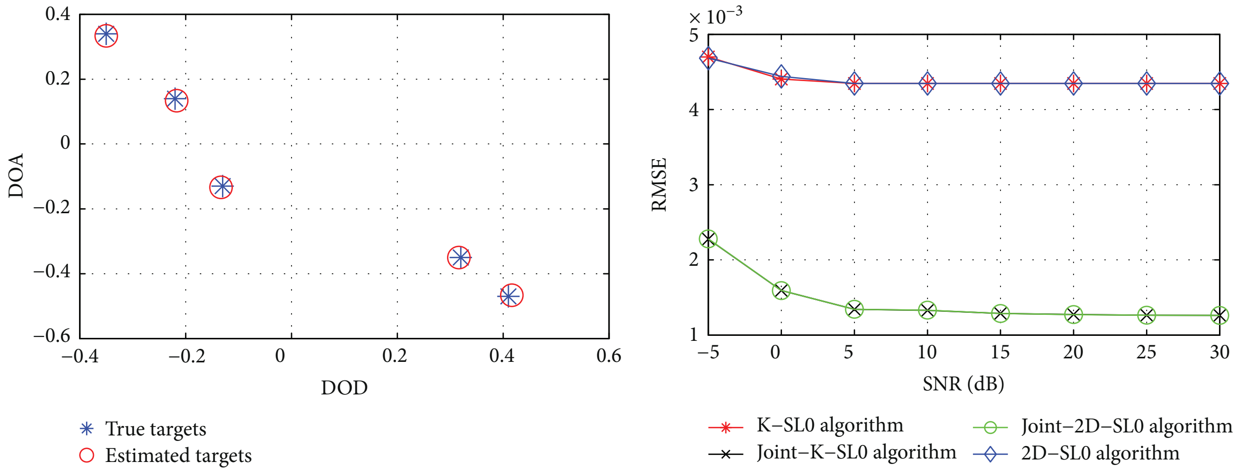
(3) Set $\begin{bmatrix} \Xi^j & \mathbf{P}_2^j \\ \mathbf{P}_1^j & \mathbf{P}_3^j \end{bmatrix} = \begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix}$.

Final answer is: $\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} = \begin{bmatrix} \Xi^j & \mathbf{P}_2^j \\ \mathbf{P}_1^j & \mathbf{P}_3^j \end{bmatrix}$.

ALGORITHM 1: The Joint-2D-SL0 algorithm.

Remark 2. Steepest ascent consists of iterations of the form $\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} \leftarrow \begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} + \mu_j \nabla G_\sigma$. Here, the step-size parameters μ_j should be decreasing, that is, for smaller values of σ, smaller values of μ_j should be applied. Note that instead of μ_j only a constant μ appeared. The reason is that by letting $\mu_j = \mu\sigma^2$ for some constant μ, we have $\begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} \leftarrow \begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} + \mu\sigma^2 \nabla G_\sigma = \begin{bmatrix} \Xi & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_3 \end{bmatrix} + \mu\delta$.

$$\delta = \sigma^2 \nabla G_\sigma = \begin{bmatrix} -\Xi_{i,j}\exp\left(-\frac{\Xi_{i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2}{\sigma^2}\right) & \cdots & -\mathbf{P}_{2i,j}\exp\left(-\frac{\Xi_{i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2}{\sigma^2}\right) \\ \vdots & \ddots & \vdots \\ -\mathbf{P}_{1i,j}\exp\left(-\frac{\Xi_{i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2}{\sigma^2}\right) & \cdots & -\mathbf{P}_{3i,j}\exp\left(-\frac{\Xi_{i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2}{\sigma^2}\right) \end{bmatrix}. \quad (14)$$



(a) Algorithm correctness verification

(b) Estimation accuracy comparison of the proposed algorithm

Figure 1: The estimation performance of the proposed algorithms.

Remark 3. The initial value our algorithm is the minimum L_2 norm solution of $\mathbf{Y}=\begin{bmatrix}\mathbf{\Phi}_R & \mathbf{\Phi}_R{}'\end{bmatrix}\begin{bmatrix}\mathbf{\Xi}^l & \mathbf{P}_2{}^l \\ \mathbf{P}_1{}^l & \mathbf{P}_3{}^l\end{bmatrix}\begin{bmatrix}\mathbf{\Phi}_T & \mathbf{\Phi}_T{}'\end{bmatrix}^T + \mathbf{Z}$, which corresponds to $\sigma \to \infty$. And the specific proof can refer to [15].

Remark 4. Having initiated the algorithm with the minimum L_2 norm solution (which corresponds to $\sigma \to \infty$), the next value for σ (i.e., σ_1) may be about 6 to 12 times of the maximum absolute value of the obtained sources. Here, we select $\sigma_1 > 8\max_{i,j}|\mathbf{W}_{i,j}|$. To see the reason, if we take, for example, $\sigma_1 > 8\max_{i,j}|\mathbf{W}_{i,j}|$, then $\exp((\mathbf{\Xi}_{1i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2)/\sigma^2) > 0.93 \approx 1$ for all i and j. And it shows that this value of σ acts virtually like infinity for all the values of $(\mathbf{\Xi}_{1i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2)^{0.5}$.

Remark 5. The smallest value of σ should be about three to six times of (a rough estimation of) the standard deviation of this noise. This is because, while σ is in this range, the cost function treats small (noisy) samples as zeros (i.e., for which $\exp((\mathbf{\Xi}_{1i,j}{}^2 + \mathbf{P}_{1i,j}{}^2 + \mathbf{P}_{2i,j}{}^2 + \mathbf{P}_{3i,j}{}^2)/\sigma^2) \approx 1$). However, below this range, the algorithm tries to "learn" these noisy values and moves away from the true answer.

Remark 6. The sequence of σ is always chosen as a decreasing geometrical sequence $\sigma_j = c\sigma_{j-1}$, $j \geq 2$, which is determined by the first and last elements, σ_1 and σ_J, and the scale factor c. In our simulation, for increasing the speed, we set $J = 100$. So $c = (\sigma_J/\sigma_1)^{1/J}$.

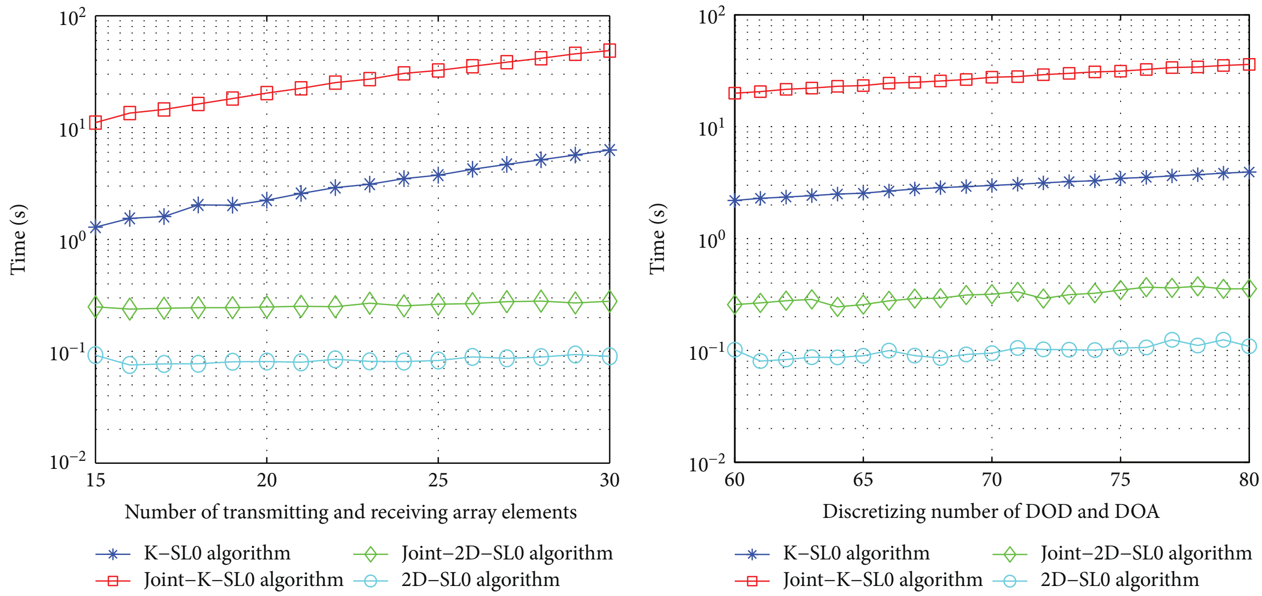
4. Simulation Results

In this section, we conduct several simulation experiments to verify the performance of the proposed model and algorithm.

In the first simulation, we show the correctness of the proposed Joint-2D-SL0 algorithm. Assume there are 5 targets, their normalized DOA and DOD are [0.34, 0.14, −0.13, −0.35, and −0.47] and [−0.35, −0.22, −0.13, 0.32, and 0.41], respectively. There are 20 transmitters and 20 receivers. The grid number of DOD and DOA is 60 and 60, respectively. The noise is white Gaussian noise. And the SNR is 20 dB. The input parameters are $\sigma_{\min} = 0.1$, $J = 100$, $\mu = 2$, and $L = 3$. The result is shown in Figure 1(a). We can see that the proposed Joint-2D-SL0 algorithm can accurately estimate the parameters of the target which verifies the correctness of the proposed algorithm.

Then, we compare the estimation accuracy of the proposed Joint-2D-SL0 algorithm with the estimation accuracy of Joint-K-SL0 algorithm (solving (5) using Kronecker compressive sensing method) and K-SL0 algorithm (solving (2) using Kronecker compressive sensing method) and 2D-SL0 algorithm (solving (2) using the method in [9]). The SNR varies from −5 dB to 30 dB. And the other parameters remain the same. 200 times simulation is conducted in each SNR. The simulation result is in Figure 1(b). We can see that the estimation error of the proposed model decreases significantly with the increase of SNR while the estimated error of the traditional model does not decrease when the SNR is bigger than 0 dB. That is because when SNR > 0 dB, the estimation error of traditional model is mainly affected by the grid mismatch problem. And the increase of SNR will not bring accuracy improvement. So the proposed off-grid model has a better performance than the model without considering grid mismatch problem which proves the validity of the proposed model. From Figure 1(b), we can see that the estimation accuracy of standard 2D-SL0 is similar with the estimation accuracy of K-SL0 algorithm. And their estimation accuracy is much lower than the estimation accuracy of Joint-2D-SL0. We also can see that the proposed Joint-2D-SL0 algorithm has a similar performance with Joint-K-SL0 algorithm which verifies the effectiveness of the proposed algorithm.

In the third simulation, we compare the running time of these algorithms. For the convenience of comparison, we conduct two kinds of contrast experiment. In the first



(a) Grid number of DOD and DOA both set to be 60

(b) The number of transmitter and receiver array elements both set to be 20

FIGURE 2: Runtime comparison of algorithm.

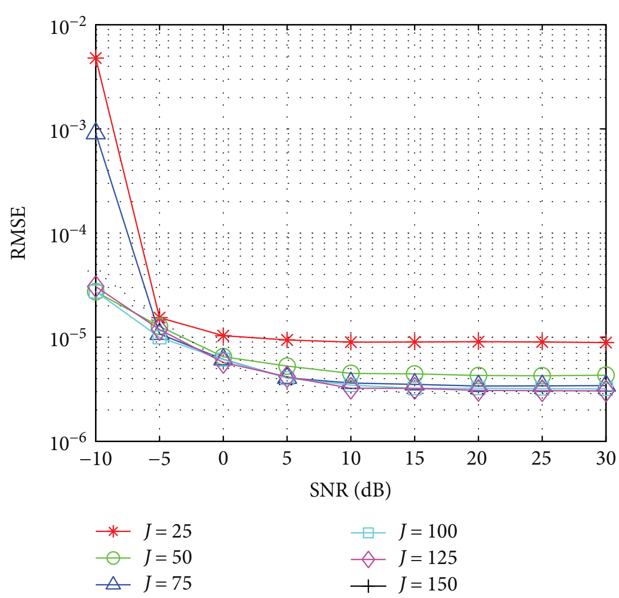


FIGURE 3: Comparison of different J.

comparative experiment, we set the grid number of DOD and DOA both to be 60, the number of transmitter and receiver elements is set to the same value which varies from 15 to 30. And the result is shown in Figure 2(a). In the second comparative experiment, the number of transmitter and receiver elements is both set to be 20 while the grid number of DOD and DOA is set to the same value which varies from 60 to 80. And the result is shown in Figure 2(b). The other simulation parameters are the same with the first simulation. We can see that the runtime of the proposed 2D algorithm is the fastest algorithm compared with the K-SL0 and Joint-K-SL0 algorithms in the two contrast simulation. And the runtime increasing tendency of the proposed 2D method is the lowest compared with the runtime increasing tendency of the other two algorithms which proves the efficiency of the proposed Joint-2D-SL0 algorithm. From Figure 2, we can see that the runtime of standard 2D-SL0 is shorter than that of Joint-2D-SL0. But the estimation accuracy of standard 2D-SL0 is much lower than that of Joint-2D-SL0, so we can say that the proposed Joint-2D-SL0 algorithm can get high estimation accuracy with acceptable computation complexity.

In the fourth experiment, to show the affection of the number of iterations, we set $J = 25$, 50, 75, 100, 125, and 150, respectively. The other parameters are same with the second experiment. And the result is shown in Figure 3. From this figure, we can see that the estimation accuracy improved with the increase of J. However, when J reaches to a certain number, the estimation accuracy remains constant and does not increase anymore. So, the optimal choice of J depends on the application. When SNR is the essential criterion, J should be chosen large, but this will result in a higher computational cost. Therefore, the choice of J is a trade-off between SNR and computational cost.

5. Conclusion

In this paper, we propose the joint sparse matrix reconstruction model based on the one-order Taylor expansion and it can overcome the grid mismatch problem efficiently. Then, we put forward the Joint-2D-SL0 algorithm to solve the joint sparse matrix reconstruction problem. Our algorithm can get high estimation accuracy with acceptable computational complexity. Simulation experiences verify the effectiveness of the proposed model and algorithm.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61571459, 61372166, and 61701526.

References

[1] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[3] E. J. Candes and J. Romberg, "Quantitative robust uncertainty principles and optimally sparse decompositions," *Foundations of Computational Mathematics*, vol. 6, no. 2, pp. 227–254, 2006.

[4] Z. Dornoosh and A. K. Ashraf, "Low rank and sparse matrix reconstruction with partial support knowledge for surveillance video processing," in *2013 IEEE International Conference on Image Processing*, pp. 335–339, Melbourne, VIC, Australia, 2013.

[5] K. Anastasios and C. Volkan, "MATRIX ALPS: accelerated low rank and sparse matrix reconstruction," in *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 185–188, Ann Arbor, MI, USA, 2012.

[6] X. Y. Cui, J. Z. Huang, S. T. Zhang, and D. N. Metaxas, "Background subtraction using low rank and group sparsity constraints," in *European Conference on Computer Vision*, pp. 612–625, Springer, Berlin, Heidelberg, 2012.

[7] Y. Liu, M. Y. Wu, and S. J. Wu, "Fast OMP algorithm for 2D angle estimation in MIMO radar," *Electronics Letters*, vol. 46, no. 6, p. 444, 2010.

[8] W. Feng, X. Y. He, Y. S. Zhang, and Y. D. Guo, "2D OMP algorithm for space–time parameters estimation of moving targets," *Electronics Letters*, vol. 51, no. 22, pp. 1809–1811, 2015.

[9] A. Ghaffari, M. Babaie-Zadeh, and C. Jutten, "Sparse decomposition of two dimensional signal," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3157–3160, Taipei, Taiwan, April 2009.

[10] M. J. Jahromi and M. H. Kahaei, "Two-dimensional iterative adaptive approach for sparse matrix solution," *Electronics Letters*, vol. 50, no. 1, pp. 45–47, 2014.

[11] Z. Tan and A. Nehorai, "Sparse direction of arrival estimation using co-prime arrays with off-grid targets," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 26–29, 2014.

[12] Y. Chi, L. L. Scharf, A. Pezeshki, and A. R. Calderbank, "Sensitivity to basis mismatch in compressed sensing," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2182–2195, 2011.

[13] Z. Tan, P. Yang, and A. Nehorai, "Joint sparse recovery method for compressed sensing with structured dictionary mismatches," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 4997–5008, 2014.

[14] A. Bacci, D. Staglianò, E. Giusti, S. Tomei, F. Berizzi, and M. Martorella, "Compressive sensing for interferometric inverse synthetic aperture radar applications," *IET Radar, Sonar & Navigation*, vol. 10, no. 8, pp. 1446–1457, 2016.

[15] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 289–301, 2009.