

Supplementary Materials for: Sparse Unmixing of Hyperspectral Data Using Spectral a Prior Information

Wei Tang, Zhenwei Shi, *Member, IEEE*, Ying Wu, *Senior Member, IEEE*, and Changshui Zhang, *Member, IEEE*

I. DETAILED DEDUCTION OF THE SUNSPI ALGORITHM

Now we present the SUNSPI algorithm to solve the following problem:

$$\min_{\mathbf{X}} \frac{1}{2} \|\mathbf{A}\mathbf{X} - \mathbf{Y}\|_F^2 + \lambda_S \sum_{i=1}^K \|\mathbf{x}_i\|_1 + \lambda_P \sum_{i \in \mathbf{S}/\mathbf{P}} \|\mathbf{x}^i\|_2$$

subject to: $\mathbf{X} \geq 0$. (1)

Remember we use $\|\mathbf{X}\|_1$ and $\|\mathbf{X}\|_{2,1}$ to denote $\sum_{i=1}^K \|\mathbf{x}_i\|_1$ and $\sum_{i=1}^m \|\mathbf{x}^i\|_2$, respectively. $\mathbf{H} \in R^{m \times m}$ is a diagonal matrix related to the set \mathbf{P} :

$$h_{ii} = \begin{cases} 0, & \text{if } i \in \mathbf{P} \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

where h_{ii} is the i -th diagonal element of \mathbf{H} . Thus, we have $\sum_{i \in \mathbf{S}/\mathbf{P}} \|\mathbf{x}^i\|_2 = \|\mathbf{H}\mathbf{X}\|_{2,1}$. Then the SUNSPI model in Eq. (1) can be written in the following equivalent form:

$$\min_{\mathbf{X}} \frac{1}{2} \|\mathbf{A}\mathbf{X} - \mathbf{Y}\|_F^2 + \lambda_S \|\mathbf{X}\|_1 + \lambda_P \|\mathbf{H}\mathbf{X}\|_{2,1} + l_{R^+}(\mathbf{X}) \quad (3)$$

where $l_{R^+}(\mathbf{X})$ is the indicator function: $l_{R^+}(\mathbf{X})$ is zero if $\mathbf{X} \geq 0$ is satisfied and $+\infty$ otherwise.

Wei Tang is with Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, PR China (e-mail: tangwei@sa.buaa.edu.cn).

Zhenwei Shi (Corresponding Author) is with Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, PR China, with State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, PR China and also with Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, PR China (e-mail: shizhenwei@buaa.edu.cn).

Ying Wu is with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 USA (e-mail: yingwu@eecs.northwestern.edu).

Changshui Zhang is with the State Key Laboratory on Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: zcs@mail.tsinghua.edu.cn).

The optimization problem in Eq. (3) has the following equivalent formulation:

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{V}} \frac{1}{2} \|\mathbf{V}_1 - \mathbf{Y}\|_F^2 + \lambda_S \|\mathbf{V}_2\|_1 + \lambda_P \|\mathbf{V}_3\|_{2,1} + l_{R^+}(\mathbf{V}_4) \\ \text{subject to: } \mathbf{V}_1 = \mathbf{A}\mathbf{U} \\ \mathbf{V}_2 = \mathbf{U} \\ \mathbf{V}_3 = \mathbf{H}\mathbf{U} \\ \mathbf{V}_4 = \mathbf{U} \end{aligned} \quad (4)$$

where

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \\ \mathbf{V}_4 \end{bmatrix}. \quad (5)$$

Suppose \mathbf{I} is identity matrix with proper size. We can write Eq. (4) in a more compact form:

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{V}} g(\mathbf{V}) \\ \text{subject to: } \mathbf{G}\mathbf{U} + \mathbf{B}\mathbf{V} = 0 \end{aligned} \quad (6)$$

where

$$\begin{aligned} g(\mathbf{V}) \equiv \frac{1}{2} \|\mathbf{V}_1 - \mathbf{Y}\|_F^2 + \lambda_S \|\mathbf{V}_2\|_1 + \lambda_P \|\mathbf{V}_3\|_{2,1} + l_{R^+}(\mathbf{V}_4) \\ \mathbf{G} = \begin{bmatrix} \mathbf{A} \\ \mathbf{I} \\ \mathbf{H} \\ \mathbf{I} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix}. \end{aligned} \quad (7)$$

The ADMM algorithm [1] for solving the problem in Eq. (6) is shown in Algorithm 1, where

$$\ell(\mathbf{U}, \mathbf{V}, \mathbf{D}) \equiv g(\mathbf{V}) + \frac{\mu}{2} \|\mathbf{G}\mathbf{U} + \mathbf{B}\mathbf{V} - \mathbf{D}\|_F^2 \quad (8)$$

is the augmented Lagrangian for the problem in Eq. (6). Here, $\mu > 0$ is the augmented Lagrangian penalty parameter [2], and $\mu\mathbf{D}$ denotes the Lagrange multipliers related to the constraint $\mathbf{G}\mathbf{U} + \mathbf{B}\mathbf{V} = 0$. In each iteration, the ADMM algorithm sequentially minimizes ℓ with respect to \mathbf{U} and \mathbf{V} , and then updates the Lagrange multipliers.

Remember we update μ by keeping the ratio between the ADMM primal residual norm and dual residual norm within a given positive interval, as they both converge to zero. Here we make use of the KKT conditions [2] to derive the primal and dual residuals for ADMM. First, the primal variables must be feasible, which leads to the condition:

$$\mathbf{G}\mathbf{U}^* + \mathbf{B}\mathbf{V}^* = 0 \quad (9)$$

Algorithm 1 Alternating direction method of multipliers

(ADMM) pseudocode for solving the problem in Eq. (6)

1: **Initialization:** set $k = 0$, choose $\mu > 0$, $\mathbf{U}^0, \mathbf{V}^0, \mathbf{D}^0$

2: **repeat:**

3: $\mathbf{U}^{(k+1)} \leftarrow \arg \min_{\mathbf{U}} \ell(\mathbf{U}, \mathbf{V}^{(k)}, \mathbf{D}^{(k)})$

4: $\mathbf{V}^{(k+1)} \leftarrow \arg \min_{\mathbf{V}} \ell(\mathbf{U}^{(k+1)}, \mathbf{V}, \mathbf{D}^{(k)})$

5: $\mathbf{D}^{(k+1)} \leftarrow \mathbf{D}^{(k)} - \mathbf{G}\mathbf{U}^{(k+1)} - \mathbf{B}\mathbf{V}^{(k+1)}$

6: **until** some stopping criterion is satisfied.

where \mathbf{U}^* and \mathbf{V}^* are the optimal solution of the problem in Eq. (6). Next, the dual variables should satisfy the Lagrange multiplier (or dual feasibility) condition [3]:

$$0 \in \partial g(\mathbf{V}^*) - \mathbf{B}^T \lambda^* \quad (10)$$

$$0 = -\mathbf{G}^T \lambda^* \quad (11)$$

where $\partial g(\mathbf{V}^*)$ means the subdifferential of a convex function g at \mathbf{V}^* , $\lambda \equiv \mu \mathbf{D}$ is the Lagrange multipliers for the problem in Eq. (6).

From the optimality condition for Step 4 of Algorithm 1, we have

$$0 \in \partial g(\mathbf{V}^{(k+1)}) + \mu \mathbf{B}^T (\mathbf{G}\mathbf{U}^{(k+1)} + \mathbf{B}\mathbf{V}^{(k+1)} - \mathbf{D}^{(k)}) \quad (12)$$

$$= \partial g(\mathbf{V}^{(k+1)}) - \mu \mathbf{B}^T \mathbf{D}^{(k+1)} \quad (13)$$

$$= \partial g(\mathbf{V}^{(k+1)}) - \mathbf{B}^T \lambda^{(k+1)} \quad (14)$$

Thus, the dual optimality condition in Eq. (10) is satisfied by $\mathbf{V}^{(k+1)}$ and $\lambda^{(k+1)}$ at the end of each iteration of Algorithm 1.

To meet the dual optimality condition in Eq. (11), we exploit the optimality condition for Step 3 of Algorithm 1:

$$0 = \mu \mathbf{G}^T (\mathbf{G}\mathbf{U}^{(k+1)} + \mathbf{B}\mathbf{V}^{(k)} - \mathbf{D}^{(k)}) \quad (15)$$

$$= -\mu \mathbf{G}^T \mathbf{D}^{(k+1)} - \mu \mathbf{G}^T \mathbf{B} (\mathbf{V}^{(k+1)} - \mathbf{V}^{(k)}) \quad (16)$$

$$= -\mathbf{G}^T \lambda^{(k+1)} - \mu \mathbf{G}^T \mathbf{B} (\mathbf{V}^{(k+1)} - \mathbf{V}^{(k)}) \quad (17)$$

Thus, after each iteration of Algorithm 1, we have

$$\mu \mathbf{G}^T \mathbf{B} (\mathbf{V}^{(k+1)} - \mathbf{V}^{(k)}) = -\mathbf{G}^T \lambda^{(k+1)} \quad (18)$$

Finally, the primal residual ($r^{(k)}$) and dual residual ($d^{(k)}$) which can measure how well the iterates of Algorithm 1 satisfy the KKT conditions can be defined as [3]:

$$r^{(k)} = \mathbf{G}\mathbf{U}^{(k)} + \mathbf{B}\mathbf{V}^{(k)} \quad (19)$$

$$d^{(k)} = \mu \mathbf{G}^T \mathbf{B} (\mathbf{V}^{(k)} - \mathbf{V}^{(k-1)}) \quad (20)$$

Now, we detail the SUnSPI algorithm. We first expand the augmented Lagrangian in Eq. (8):

$$\begin{aligned}
& \ell(\mathbf{U}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3, \mathbf{V}_4, \mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \mathbf{D}_4) \\
&= \frac{1}{2} \|\mathbf{V}_1 - \mathbf{Y}\|_F^2 + \lambda_S \|\mathbf{V}_2\|_1 + \lambda_P \|\mathbf{V}_3\|_{2,1} + l_{R+}(\mathbf{V}_4) \\
&+ \frac{\mu}{2} \|\mathbf{A}\mathbf{U} - \mathbf{V}_1 - \mathbf{D}_1\|_F^2 + \frac{\mu}{2} \|\mathbf{U} - \mathbf{V}_2 - \mathbf{D}_2\|_F^2 \\
&+ \frac{\mu}{2} \|\mathbf{H}\mathbf{U} - \mathbf{V}_3 - \mathbf{D}_3\|_F^2 + \frac{\mu}{2} \|\mathbf{U} - \mathbf{V}_4 - \mathbf{D}_4\|_F^2.
\end{aligned} \tag{21}$$

In each iteration of the ADMM scheme, we should sequentially minimize the function ℓ in Eq. (21) with respect to \mathbf{U} , \mathbf{V}_1 , \mathbf{V}_2 , \mathbf{V}_3 and \mathbf{V}_4 , and then update the Lagrange multipliers.

We first run an optimization over the variable \mathbf{U} . By ignoring the terms in the objective function in Eq. (21) that do not contain variable \mathbf{U} , we can get the reduced optimization problem:

$$\begin{aligned}
\mathbf{U}^{(k+1)} \leftarrow \arg \min_{\mathbf{U}} & \frac{\mu}{2} \|\mathbf{A}\mathbf{U} - \mathbf{V}_1^{(k)} - \mathbf{D}_1^{(k)}\|_F^2 + \\
& \frac{\mu}{2} \|\mathbf{U} - \mathbf{V}_2^{(k)} - \mathbf{D}_2^{(k)}\|_F^2 + \frac{\mu}{2} \|\mathbf{H}\mathbf{U} - \mathbf{V}_3^{(k)} - \mathbf{D}_3^{(k)}\|_F^2 \\
& + \frac{\mu}{2} \|\mathbf{U} - \mathbf{V}_4^{(k)} - \mathbf{D}_4^{(k)}\|_F^2
\end{aligned} \tag{22}$$

which has closed form solution:

$$\begin{aligned}
\mathbf{U}^{(k+1)} \leftarrow & (\mathbf{A}^T \mathbf{A} + 2\mathbf{I} + \mathbf{H})^{-1} [\mathbf{A}^T (\mathbf{V}_1^{(k)} + \mathbf{D}_1^{(k)}) \\
& + \mathbf{V}_2^{(k)} + \mathbf{D}_2^{(k)} + \mathbf{H}(\mathbf{V}_3^{(k)} + \mathbf{D}_3^{(k)}) + \mathbf{V}_4^{(k)} + \mathbf{D}_4^{(k)}].
\end{aligned} \tag{23}$$

Here we exploit the fact that \mathbf{H} is a diagonal matrix with diagonal elements 0 or 1, which means $\mathbf{H}^T = \mathbf{H}$ and $\mathbf{H}^T \mathbf{H} = \mathbf{H}$.

Then we turn to compute the values of variables \mathbf{V}_1 , \mathbf{V}_2 , \mathbf{V}_3 and \mathbf{V}_4 at each iteration. To update \mathbf{V}_1 , the reduced optimization problem is

$$\mathbf{V}_1^{(k+1)} \leftarrow \arg \min_{\mathbf{V}_1} \frac{1}{2} \|\mathbf{V}_1 - \mathbf{Y}\|_F^2 + \frac{\mu}{2} \|\mathbf{A}\mathbf{U}^{(k)} - \mathbf{V}_1 - \mathbf{D}_1^{(k)}\|_F^2 \tag{24}$$

whose solution is

$$\mathbf{V}_1^{(k+1)} \leftarrow \frac{1}{1 + \mu} [\mathbf{Y} + \mu(\mathbf{A}\mathbf{U}^{(k)} - \mathbf{D}_1^{(k)})] \tag{25}$$

Similarly, the reduced optimization problem for \mathbf{V}_2 is

$$\mathbf{V}_2^{(k+1)} \leftarrow \arg \min_{\mathbf{V}_2} \lambda_S \|\mathbf{V}_2\|_1 + \frac{\mu}{2} \|\mathbf{U}^{(k)} - \mathbf{V}_2 - \mathbf{D}_2^{(k)}\|_F^2 \tag{26}$$

whose solution is the *soft threshold* [4]:

$$\mathbf{V}_2^{(k+1)} \leftarrow \text{soft}(\xi_2, \frac{\lambda_S}{\mu}) \tag{27}$$

where $\xi_2 = \mathbf{U}^{(k)} - \mathbf{D}_2^{(k)}$ and $\text{soft}(\cdot, \tau)$ denotes the component-wise application of the soft-threshold function $y \mapsto \text{sign}(y) \max\{|y| - \tau, 0\}$.

To update \mathbf{V}_3 , the reduced optimization problem is

$$\mathbf{V}_3^{(k+1)} \leftarrow \arg \min_{\mathbf{V}_3} \lambda_P \|\mathbf{V}_3\|_{2,1} + \frac{\mu}{2} \|\mathbf{H}\mathbf{U}^{(k)} - \mathbf{V}_3 - \mathbf{D}_3^{(k)}\|_F^2 \quad (28)$$

whose solution is the well-known *vect-soft threshold* [5], applied independently to each row r of the update variable

$$(\mathbf{V}_3^{(k+1)})_r \leftarrow \text{vect-soft}(\xi_3^r, \frac{\lambda_P}{\mu}) \quad (29)$$

where $\xi_3 = \mathbf{H}\mathbf{U}^{(k)} - \mathbf{D}_3^{(k)}$ and $\text{vect-soft}(\cdot, \tau)$ denotes the row-wise application of the vect-soft-threshold function $\mathbf{y} \mapsto \mathbf{y} \max\{\|\mathbf{y}\|_2 - \tau, 0\} / (\max\{\|\mathbf{y}\|_2 - \tau, 0\} + \tau)$.

To compute \mathbf{V}_4 , we solve the following optimization problem

$$\mathbf{V}_4^{(k+1)} \leftarrow \arg \min_{\mathbf{V}_4} l_{R+}(\mathbf{V}_4) + \frac{\mu}{2} \|\mathbf{U}^{(k)} - \mathbf{V}_4 - \mathbf{D}_4^{(k)}\|_F^2 \quad (30)$$

whose solution is the projection of $(\mathbf{U}^{(k)} - \mathbf{D}_4^{(k)})$ onto the nonnegative orthant:

$$\mathbf{V}_4^{(k+1)} \leftarrow \max\{\mathbf{U}^{(k)} - \mathbf{D}_4^{(k)}, 0\}. \quad (31)$$

After updating \mathbf{U} and \mathbf{V} in each iteration, we should update the Lagrange multipliers. The whole process of SUnSPI algorithm is shown in Algorithm 2.

Algorithm 2 Pseudocode of the SUnSPI algorithm

1: **Initialization:**

2: set $k = 0$, choose $\mu > 0$, $\mathbf{U}^0, \mathbf{V}_1^0, \mathbf{V}_2^0, \mathbf{V}_3^0, \mathbf{V}_4^0, \mathbf{D}_1^0, \mathbf{D}_2^0, \mathbf{D}_3^0, \mathbf{D}_4^0$

3: **repeat:**

4: Compute $\mathbf{U}^{(k+1)}$ via Eq. (23)

5: Compute $\mathbf{V}_1^{(k+1)}$ via Eq. (25)

6: Compute $\mathbf{V}_2^{(k+1)}$ via Eq. (27)

7: Compute $\mathbf{V}_3^{(k+1)}$ via Eq. (29)

8: Compute $\mathbf{V}_4^{(k+1)}$ via Eq. (31)

9: $\mathbf{D}_1^{(k+1)} \leftarrow \mathbf{D}_1^{(k)} - \mathbf{A}\mathbf{U}^{(k+1)} + \mathbf{V}_1^{(k+1)}$

10: $\mathbf{D}_2^{(k+1)} \leftarrow \mathbf{D}_2^{(k)} - \mathbf{U}^{(k+1)} + \mathbf{V}_2^{(k+1)}$

11: $\mathbf{D}_3^{(k+1)} \leftarrow \mathbf{D}_3^{(k)} - \mathbf{H}\mathbf{U}^{(k+1)} + \mathbf{V}_3^{(k+1)}$

12: $\mathbf{D}_4^{(k+1)} \leftarrow \mathbf{D}_4^{(k)} - \mathbf{U}^{(k+1)} + \mathbf{V}_4^{(k+1)}$

13: Update iteration: $k \leftarrow k + 1$

14: **until** some stopping criterion is satisfied.

II. MORE EXPERIMENTAL RESULTS

To further prove the effectiveness of the proposed algorithm, we also conduct experiments on another frequently-used synthetic data set.

The fourth synthetic data (SD4) are created as follows [6–9]:

TABLE I
RMSES OBTAINED BY DIFFERENT ALGORITHMS USING \mathbf{A}_1 ON SD4 CORRUPTED BY WHITE NOISE

	SNR (dB)	SUnSAL	CLSUnSAL	NCLS-SPI(1)	NCLS-SPI(2)	SUnSPI(0)	SUnSPI(1)	SUnSPI(2)
SD4 ($k_1 = 3$)	20	0.0713	0.0390	0.0214	0.0202	0.0390	0.0214	0.0200
	30	0.0242	0.0174	0.0092	0.0089	0.0161	0.0092	0.0085
	40	0.0075	0.0076	0.0046	0.0043	0.0068	0.0046	0.0040
SD4 ($k_2 = 6$)	20	0.0682	0.0593	0.0569	0.0356	0.0534	0.0473	0.0346
	30	0.0226	0.0217	0.0201	0.0126	0.0193	0.0175	0.0122
	40	0.0075	0.0072	0.0072	0.0053	0.0068	0.0063	0.0050
SD4 ($k_3 = 9$)	20	0.0643	0.0638	0.0549	0.0520	0.0581	0.0510	0.0494
	30	0.0372	0.0366	0.0323	0.0172	0.0350	0.0314	0.0168
	40	0.0266	0.0272	0.0249	0.0084	0.0258	0.0249	0.0084

- 1) Divide the scene, whose size is $z^2 \times z^2$ ($z = 8$), into $z \times z$ regions. P endmembers ($P = 3, 6, 9$) are selected randomly from \mathbf{A}_1 to constitute the endmember class. Initialize each region with the same type of ground cover, randomly selected from the endmember class. The size of spectral signatures matrix \mathbf{W} is $L \times P$ ($L = 224$).
- 2) Generate mixed pixels through a simple $(z + 1) \times (z + 1)$ spatial low-pass filter.
- 3) Replace all the pixels in which the abundance of a single endmember is larger than 70% with a mixture made up of this endmember and its next endmember (the abundances of the two endmembers both equal 50%) so as to further remove pure pixels and represent the sparseness of abundances at the same time; After these three steps, we obtain the distribution of P endmembers in the scene and the abundance values are stored in \mathbf{H} with a size of $P \times K$ ($K = z^2 \times z^2$).
- 4) Use linear spectral mixing model $\mathbf{Y} = \mathbf{W} \times \mathbf{H}$ to generate hyperspectral data, add Gaussian white noise or correlated noise with specific SNR at the same time. The size of hyperspectral data \mathbf{Y} is $L \times K$.

Tabs. I and II show the results obtained by the considered sparse unmixing algorithms on SD4 in situations of different SNRs, different endmember numbers and different noise types. We can see that the performances of different algorithms on SD4 are similar to those on SD1. We can observe that in all the cases SUnSPI(2) and SUnSPI(1) behave better than SUnSPI(0), SUnSAL and CLSUnSAL; SUnSPI(2) outperforms SUnSPI(1); NCLS-SPI(1) and NCLS-SPI(2) outperform CLSUnSAL; NCLS-SPI(2) behaves better than NCLS-SPI(1). All these observations indicate that the spectral a priori information is beneficial to sparse unmixing and more such information results in better estimation of the abundances. We can also find that when the noise is weak or endmember number is small, the difference between the performances of NCLS-SPI and SUnSPI is slight. Thus, in these cases, setting $\lambda_S = 0$ for SUnSPI can achieve satisfactory results.

Fig. 1 shows the true abundance maps and abundance maps obtained by different algorithms using \mathbf{A}_1 on SD4 with 20 dB white noise when the endmember number is 6. In this case, SUnSPI(1) and NCLS-SPI(1) know that the third and fourth actual endmembers exist in the data; SUnSPI(2) and NCLS-SPI(2) know that the first, third, fourth and fifth actual endmembers are present in the hyperspectral scene. By visual comparison, it is obvious

TABLE II
RMSEs OBTAINED BY DIFFERENT ALGORITHMS USING \mathbf{A}_1 ON SD4 CORRUPTED BY CORRELATED NOISE

	SNR (dB)	SUnSAL	CLSunSAL	NCLS-SPI(1)	NCLS-SPI(2)	SUnSPI(0)	SUnSPI(1)	SUnSPI(2)
SD4 ($k_1 = 3$)	20	0.0727	0.0770	0.0244	0.0242	0.0547	0.0243	0.0240
	30	0.0213	0.0238	0.0092	0.0092	0.0212	0.0092	0.0092
	40	0.0076	0.0085	0.0041	0.0035	0.0071	0.0039	0.0032
SD4 ($k_2 = 6$)	20	0.0728	0.0847	0.0794	0.0435	0.0644	0.0537	0.0417
	30	0.0229	0.0289	0.0281	0.0152	0.0213	0.0187	0.0148
	40	0.0079	0.0089	0.0087	0.0051	0.0077	0.0071	0.0051
SD4 ($k_3 = 9$)	20	0.0694	0.0819	0.0668	0.0638	0.0655	0.0603	0.0588
	30	0.0380	0.0418	0.0362	0.0213	0.0374	0.0346	0.0213
	40	0.0270	0.0309	0.0263	0.0087	0.0267	0.0257	0.0087

that the more spectral a priori information is provided, the more the estimated abundances approximate the truth. The improvement is especially obvious for the abundance maps corresponding to the first and fourth endmembers which are very related to the other spectral signatures in the spectral library. Besides, we can also find that the improvement of abundance estimation for the prior endmembers is much more significant than that corresponding to the unknown endmembers.

REFERENCES

- [1] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, "An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems," *Image Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 681–695, 2011.
- [2] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.
- [3] T. Goldstein, B. ODonoghue, and S. Setzer, "Fast alternating direction optimization methods," *CAM report*, pp. 12–35, 2012.
- [4] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.
- [5] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, "Sparse reconstruction by separable approximation," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2479–2493, 2009.
- [6] L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 3, pp. 765–777, 2007.
- [7] S. Jia and Y. Qian, "Constrained nonnegative matrix factorization for hyperspectral unmixing," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 47, no. 1, pp. 161–173, 2009.
- [8] Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly, "Hyperspectral unmixing via $l_{1/2}$ sparsity-constrained nonnegative matrix factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, pp. 4282–4297, 2011.
- [9] W. Tang, Z. Shi, and Z. An, "Nonnegative matrix factorization for hyperspectral unmixing using prior

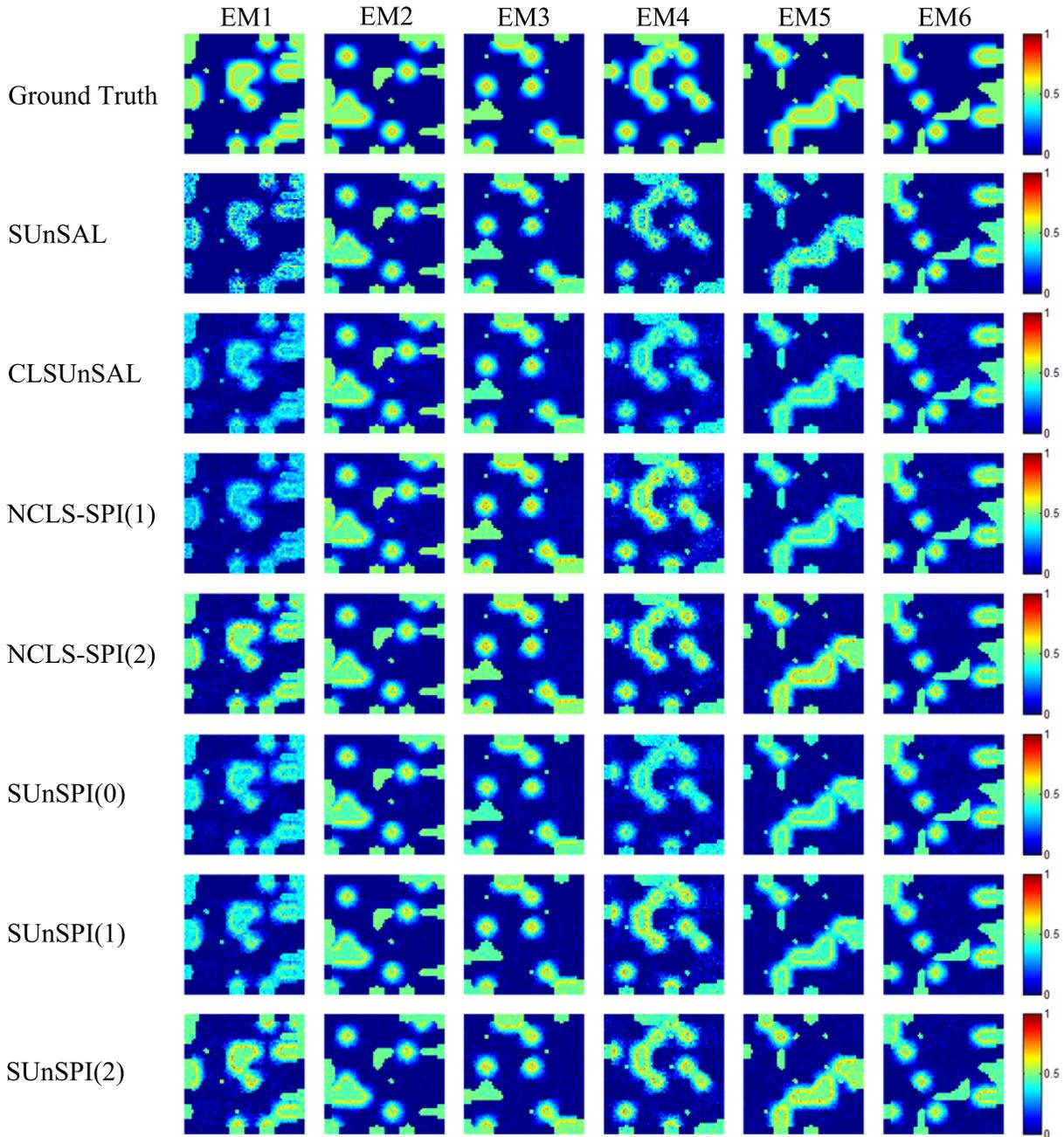


Fig. 1. Comparison of abundance maps obtained by different algorithms using A_1 on SD4 with 20 dB white noise when the endmember number is 6. From top row to bottom row are true abundance maps, abundance maps obtained by SUnSAL, abundance maps obtained by CLSUnSAL, abundance maps obtained by NCLS-SPI(1), abundance maps obtained by NCLS-SPI(2), abundance maps obtained by SUnSPI(0), abundance maps obtained by SUnSPI(1) and abundance maps obtained by SUnSPI(2), respectively.

knowledge of spectral signatures,” *Optical Engineering*, vol. 51, no. 8, pp. 087001–1–087001–10, 2012.
[Online]. Available: <http://dx.doi.org/10.1117/1.OE.51.8.087001>