Supplementary Material of "A Novel Sparsity Measure for Tensor Recovery"

Qian Zhao[1,2] Deyu Meng[1,2] Xu Kong[3] Qi Xie[1,2] Wenfei Cao[1,2] Yao Wang[1,2] Zongben Xu[1,2]
[1]School of Mathematics and Statistics, Xi'an Jiaotong University
[2]Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University
timmy.zhaoqian@gmail.com dymeng@mail.xjtu.edu.cn xq.liwu@stu.xjtu.edu.cn
{caowenf2015, yao.s.wang}@gmail.com zbxu@mail.xjtu.edu.cn
[3]School of Mathematical Sciences, Liaocheng University
xu.kong@hotmail.com

Abstract

In this supplementary material, we present details of the algorithms for solving the proposed tensor recovery models with folded-concave relaxations, and present more experimental results.

1. Solving tensor tecovery models with folded-concave relaxations

In the maintext, we have described how to solve the proposed tensor recovery models with the trace norm relaxation. Here, we present the strategy for solving the models with folded-concave relaxations.

1.1. Locally linear approximation for the folded-concave penalties

First, we introduce the idea of the locally linear approximation (LLA) for the folded-concave penalties, which is adopted from [10] and [6].

It is easy to verify that the following result holds:

Proposition 1 ([6]) *Let $f(x)$ be a concave function on $(0,\infty)$. Then*

$$f(x) \leq f(x_0) + f'(x_0)(x - x_0), \tag{1}$$

with equality if $x = x_0$.

Apply this proposition to the folded-concave penalties, we can get the LLAs for them:

$$\psi_{\text{mcp}}(t) \leq \psi_{\text{mcp}}(t_0) + \psi'_{\text{mcp}}(t_0)(t - t_0) \triangleq \widehat{\psi}_{\text{mcp}}(t|t_0), \tag{2}$$

and

$$\psi_{\text{scad}}(t) \leq \psi_{\text{scad}}(t_0) + \psi'_{\text{scad}}(t_0)(t - t_0) \triangleq \widehat{\psi}_{\text{scad}}(t|t_0), \tag{3}$$

where

$$\psi'_{\text{mcp}}(t_0) = \left(\lambda - \frac{t_0}{a}\right)_+,$$

with $(x)_+ = \max\{x, 0\}$, and

$$\psi'_{\text{scad}}(t_0) = \lambda I\{t_0 \leq \lambda\} + \frac{(a\lambda - t_0)_+}{a-1} I\{t_0 > \lambda\},$$

with $I\{A\}$ being the indicator function which equals 1 if A is true and 0 otherwise.

Using these LLAs, we can solve the original models with a two-stage approach, as discussed in the next subsection.

1.2. Two-stage algorithms for solving the models with folded-concave relaxations

We mainly discuss the algorithm for RP-TC$_{\text{mcp}}$ in the maintext, and the idea can be easily generalized to other models.

Recall the RP-TC$_{\text{mcp}}$ model:

$$\min_{\mathcal{X}} \frac{1}{C_2} S_{\text{mcp}}(\mathcal{X}), \quad s.t.\ \mathcal{X}_\Omega = \mathcal{T}_\Omega, \tag{4}$$

where

$$\begin{aligned} S_{\text{mcp}}(\mathcal{X}) &= \prod\nolimits_{n=1}^{d} P_{\text{mcp}}\left(\mathbf{X}_{(n)}\right) \\ &= \prod\nolimits_{n=1}^{d} \left(\sum\nolimits_i \psi_{\text{mcp}}(\sigma_{(n)i})\right), \end{aligned} \tag{5}$$

with $\sigma_{(n)i}$ being the ith singular value of $\mathbf{X}_{(n)}$. Now assuming that we have an initial estimation for $\mathcal{X}$, denoted as $\mathcal{X}^0$, we can apply (2) to $\psi_{\text{mcp}}(\sigma_{(n)i})$ to obtain

$$\widehat{\psi}_{\text{mcp}}(\sigma_{(n)i}|\sigma^0_{(n)i}) = \psi_{\text{mcp}}(\sigma^0_{(n)i}) + \psi'_{\text{mcp}}(\sigma^0_{(n)i})(\sigma_{(n)i} - \sigma^0_{(n)i}),$$

where $\sigma^0_{(n)i}$ is the ith singular value of the unfolding matrix $\mathbf{X}^0_{(n)}$ along the nth model of $\mathcal{X}^0$. Then we can replace ψ_{mcp}

with $\widehat{\psi}_{\text{mcp}}$, and approximate $S_{\text{mcp}}(\mathcal{X})$ as follows:

$$\widehat{S}_{\text{mcp}}(\mathcal{X}|\mathcal{X}^0) = \prod\nolimits_{n=1}^{d} \widehat{P}_{\text{mcp}}\left(\mathbf{X}_{(n)}|\mathbf{X}^0_{(n)}\right), \tag{6}$$

where

$$\widehat{P}_{\text{mcp}}\left(\mathbf{X}_{(n)}|\mathbf{X}^0_{(n)}\right) = \sum\nolimits_i \widehat{\psi}_{\text{mcp}}(\sigma_{(n)i}|\sigma^0_{(n)i}).$$

From the above discussion, we can give a two-stage algorithm for solving (4). In the first stage, we solve RP-TC$_{\text{trace}}$ to get $\mathcal{X}^0$, and then in the second stage, we solve the following problem:

$$\min_{\mathcal{X}} \frac{1}{C_2}\widehat{S}_{\text{mcp}}(\mathcal{X}|\mathcal{X}^0), \quad s.t.\ \mathcal{X}_\Omega = \mathcal{T}_\Omega. \tag{7}$$

Then the result can be used as $\mathcal{X}^0$ to solve (7) again in an iterative way. However, Zou and Li [10] showed that one-step re-estimation already can provide good result in sparse linear regression. Therefore, we also use this one-step strategy which is empirically effective in our experiments.

Now we discuss how to solve (7). We use the similar ADMM as for solving RP-TC$_{\text{trace}}$, which has been discussed in the maintext. First, we introduce d auxiliary tensors $\mathcal{M}_n$ $(1 \le n \le d)$ and reformulate (7) as follows:

$$\begin{aligned} &\min_{\mathcal{X},\mathcal{M}_1,\cdots,\mathcal{M}_d} \frac{1}{C_2}\prod\nolimits_{n=1}^{d} \widehat{P}_{\text{mcp}}\left(\mathbf{X}_{(n)}|\mathbf{X}^0_{(n)}\right) \\ &s.t.,\ \mathcal{X}_\Omega = \mathcal{M}_\Omega,\ \mathcal{X} = \mathcal{M}_n,\ 1 \le n \le d. \end{aligned} \tag{8}$$

The augmented Lagrangian function for (8) is

$$\begin{aligned} &L_\rho(\mathcal{X},\mathcal{M}_1,\cdots,\mathcal{M}_d,\mathcal{Y}_1,\cdots,\mathcal{Y}_d) \\ &= \frac{1}{C_2}\prod_{n=1}^{d}\widehat{P}_{\text{mcp}}\left(\mathbf{M}_{(n)}|\mathbf{X}^0_{(n)}\right) + \sum_{n=1}^{d}\langle \mathcal{X}-\mathcal{M}_n, \mathcal{Y}_n\rangle \\ &\quad + \frac{\rho}{2}\sum_{n=1}^{d}\|\mathcal{X}-\mathcal{M}_n\|_F^2, \end{aligned}$$

where $\mathcal{Y}_n$ $(1 \le n \le d)$ are the Lagrange multipliers and ρ is a positive scalar. Then the problem can be solved within the ADMM framework.

With $\mathcal{M}_l$ $(l \ne n)$ fixed, $\mathcal{M}_n$ can be updated by solving the following problem:

$$\min_{\mathcal{M}_n} L_\rho(\mathcal{X},\mathcal{M}_1,\cdots,\mathcal{M}_d,\mathcal{Y}_1,\cdots,\mathcal{Y}_d), \tag{9}$$

which, with some algebra, is equivalent to

$$\min_{\mathcal{M}_n} \alpha_n \widehat{P}_{\text{mcp}}\Big(\mathbf{M}_{n(n)}|\mathbf{X}^0_{(n)}\Big) + \langle \mathcal{X}-\mathcal{M}_n, \mathcal{Y}_n\rangle + \frac{\rho}{2}\|\mathcal{X}-\mathcal{M}_n\|_F^2, \tag{10}$$

where

$$\alpha_n = \frac{1}{C_2}\prod\nolimits_{l=1,l\ne n}^{d}\widehat{P}_{\text{mcp}}\Big(\mathbf{M}_{l(l)}|\mathbf{X}^0_{(l)}\Big),\ 1 \le n \le d.$$

Note that

$$\begin{aligned} \widehat{P}_{\text{mcp}}\left(\mathbf{M}_{n(n)}|\mathbf{X}^0_{(n)}\right) &= \sum\nolimits_k \widehat{\psi}_{\text{mcp}}(\sigma_{(n)k}|\sigma^0_{(n)k}) \\ &= \sum\nolimits_i \psi_{\text{mcp}}(\sigma^0_{(n)k}) + \psi'_{\text{mcp}}(\sigma^0_{(n)k})(\sigma_{(n)k} - \sigma^0_{(n)k}), \end{aligned}$$

where $\sigma_{(n)k}$ is the kth singular value of $\mathbf{M}_{n(n)}$, and then Eq. (10), by discarding constants irrelevant to $\sigma_{(n)k}$s, can be further reduced to

$$\min_{\mathcal{M}_n} \alpha_n \sum\nolimits_k \psi'_{\text{mcp}}(\sigma^0_{(n)k})\sigma_{(n)k} + \langle \mathcal{X}-\mathcal{M}_n, \mathcal{Y}_n\rangle + \frac{\rho}{2}\|\mathcal{X}-\mathcal{M}_n\|_F^2, \tag{11}$$

which is intrinsically a weighted nuclear (trace) norm minimization problem [2, 7]. Since $\sigma^0_{(n)k}$ $(k = 1, 2, \ldots, r_n)$ are of non-increasing order, and $\psi'_{\text{mcp}}(\cdot)$ is non-decreasing on $(0, \infty)$, then $\psi'_{\text{mcp}}(\sigma^0_{(n)k})$ $(k = 1, 2, \ldots, r_n)$ are of non-decreasing order. Therefore, according to [7], (11) can be analytically solved by

$$\mathcal{M}_n = \text{fold}_n\left(\widehat{\text{D}}_{\frac{\alpha_n}{\rho},\mathbf{v}_n}\Big(\text{unfold}_n\big(\mathcal{X} + \frac{1}{\rho}\mathcal{Y}_n\big)\Big)\right), \tag{12}$$

where $\mathbf{v}_n = \Big(\psi'_{\text{mcp}}(\sigma^0_{(n)1}), \psi'_{\text{mcp}}(\sigma^0_{(n)2}), \ldots, \psi'_{\text{mcp}}(\sigma^0_{(n)r_n})\Big)^T$, and $\widehat{\text{D}}_{\tau,\mathbf{w}}(\mathbf{X}) = \mathbf{U}\mathbf{\Sigma}_{\tau,\mathbf{w}}\mathbf{V}^T$ is the generalized shrinkage operator with $\mathbf{\Sigma}_{\tau,\mathbf{w}} = \text{diag}(\max(\sigma_i - \tau w_i, 0))$ (σ_is are the singular values of $\mathbf{X}$, and w_is are elements of $\mathbf{w}$).

Similarly, with $\mathcal{M}_n$s fixed, $\mathcal{X}$ can be updated by solving

$$\min_{\mathcal{X}} L_\rho(\mathcal{X},\mathcal{M}_1,\cdots,\mathcal{M}_d,\mathcal{Y}_1,\cdots,\mathcal{Y}_d), \tag{13}$$

which also has the closed-form solution:

$$\mathcal{X} = \frac{1}{\rho d}\sum\nolimits_{n=1}^{d}(\rho\mathcal{M}_n - \mathcal{Y}_n). \tag{14}$$

Then the Lagrangian multipliers are updated by

$$\mathcal{Y}_n := \mathcal{Y}_n - \rho(\mathcal{M}_n - \mathcal{X}),\ 1 \le n \le d, \tag{15}$$

and ρ is increased to $\mu\rho$ with some constant $\mu > 1$.

It is straightforward to generalize this two-stage algorithm to solving the proposed RP-TC$_{\text{scad}}$, RP-TRPCA$_{\text{mcp}}$ and RP-TRPCA$_{\text{scad}}$, and we omit them here for conciseness.

2. Additional experimental results

2.1. Detailed results on TC experiments

In the maintext, we show the averaged results of the multispectral image completion. Here, we list the results on separated images in Table 1-8 for a more comprehensive comparison. It can be seen that the proposed methods can achieve the best or second best results in all situations in terms of the 5 PQIs utilized.

2.2. Additional results on TRPCA experiments

In the maintext, we show the averaged results of the simulated multispectral image restoration. Here, we list the results on separated images in Table 9-18, and also show the visual results on 4 sample images in Fig. 1-4. We can see that the proposed methods can produce comparatively better restoration results than other competing methods, both quantitatively and visually.

References

[1] D. Goldfarb and Z. T. Qin. Robust low-rank tensor recovery: Models and algorithms. *SIAM Journal on Matrix Analysis and Applications*, 35(1):225–253, 2014.

[2] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014.

[3] Z. Lin, M. Chen, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[4] J. Liu, P. Musialski, P. Wonka, and J. P. Ye. Tensor completion for estimating missing values in visual data. *IEEE TPAMI*, 34(1):208–220, 2013.

[5] Y. Peng, D. Y. Meng, Z. B. Xu, C. Q. Gao, Y. Yang, and B. Zhang. Decomposable nonlocal tensor dictionary learning for multispectral image denoising. In *CVPR*, 2014.

[6] S. Wang, D. Liu, and Z. Zhang. Nonconvex relaxation approaches to robust matrix recovery. In *IJCAI*, 2013.

[7] Q. Xie, D. Meng, S. Gu, L. Zhang, W. Zuo, X. Feng, and Z. Xu. On the optimal solution of weighted nuclear norm minimization. *arXiv preprint arXiv:1405.6012*, 2014.

[8] Y. Xu, R. Hao, W. Yin, and Z. Su. Parallel matrix factorization for low-rank tensor completion. *arXiv preprint arXiv:1312.1254*, 2013.

[9] Z. M. Zhang, G. Ely, S. Aeron, N. Hao, and M. E. Kilmer. Novel methods for multilinear data completion and denoising based on tensor-SVD. In *CVPR*, 2014.

[10] H. Zou and R. Li. One-step sparse estimates in nonconcave penalized likelihood models. *Annals of statistics*, 36(4):1509, 2008.

Table 1. The performance comparison of 7 competing TC methods with different sampling rates on *balloons* image.

Method	10%					20%					50%				
	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM
MC-ALM [3]	28.03	0.846	0.883	168.7	0.187	33.08	0.940	0.948	94.6	0.111	41.43	0.989	0.989	36.4	0.045
HaLRTC [4]	31.41	0.926	0.940	113.9	0.127	37.61	0.975	0.975	55.9	0.068	48.11	0.997	0.997	16.7	0.025
t-SVD [9]	36.16	0.940	0.944	68.5	0.129	41.94	0.981	0.980	35.8	0.073	52.74	0.998	0.998	11.0	0.025
TMac [8]	**38.73**	0.959	0.956	**49.1**	0.094	42.66	0.980	0.978	31.3	0.064	48.96	0.995	0.995	15.2	0.030
RP-TC$_{\text{trace}}$	28.47	0.888	0.918	159.7	0.172	38.12	0.977	0.977	52.5	0.070	49.59	0.997	0.998	14.0	0.023
RP-TC$_{\text{mcp}}$	**37.99**	**0.973**	**0.971**	**53.2**	**0.085**	**45.11**	**0.994**	**0.993**	**23.7**	**0.040**	**55.06**	**0.999**	**0.999**	**7.7**	**0.015**
RP-TC$_{\text{scad}}$	37.84	**0.972**	**0.971**	54.1	**0.087**	**44.94**	**0.994**	**0.992**	**24.2**	**0.041**	**54.79**	**0.999**	**0.999**	**7.9**	**0.016**

Table 2. The performance comparison of 7 competing TC methods with different sampling rates on *beads* image.

Method	10%					20%					50%				
	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM
MC-ALM [3]	19.45	0.404	0.708	517.8	0.547	21.86	0.587	0.789	392.8	0.433	28.15	0.875	0.926	190.5	0.245
HaLRTC [4]	19.07	0.438	0.671	542.0	0.503	22.33	0.654	0.806	373.2	0.365	30.86	0.935	0.957	140.0	0.174
t-SVD [9]	23.58	0.659	0.821	320.7	0.435	28.21	0.840	0.911	190.8	0.312	38.27	0.976	0.985	63.2	0.127
TMac [8]	21.79	0.473	0.741	396.4	0.585	24.58	0.653	0.810	290.3	0.485	32.04	0.908	0.941	124.3	0.231
RP-TC$_{\text{trace}}$	18.86	0.462	0.678	555.7	0.520	22.80	0.705	0.831	354.2	0.362	34.05	0.969	0.978	97.6	0.131
RP-TC$_{\text{mcp}}$	**27.24**	**0.838**	**0.905**	**212.7**	**0.280**	**34.41**	**0.957**	**0.973**	**93.6**	**0.149**	**45.59**	**0.996**	**0.997**	**27.1**	**0.048**
RP-TC$_{\text{scad}}$	**27.51**	**0.844**	**0.908**	**206.3**	**0.274**	**34.12**	**0.954**	**0.971**	**96.6**	**0.152**	**45.51**	**0.996**	**0.997**	**27.4**	**0.048**

Table 3. The performance comparison of 7 competing TC methods with different sampling rates on *cloth* image.

Method	10%					20%					50%				
	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM
MC-ALM [3]	21.53	0.396	0.711	335.6	0.241	23.66	0.572	0.787	262.9	0.194	28.89	0.852	0.918	143.4	0.114
HaLRTC [4]	21.52	0.441	0.667	334.3	0.222	24.28	0.634	0.792	243.4	0.169	31.05	0.909	0.945	110.5	0.087
t-SVD [9]	26.37	0.729	0.863	184.4	0.166	31.37	0.890	0.942	103.7	0.104	41.07	0.980	0.989	36.9	0.041
TMac [8]	22.66	0.465	0.769	291.4	0.269	24.48	0.620	0.820	234.2	0.231	31.70	0.888	0.941	101.5	0.092
RP-TC$_{\text{trace}}$	21.36	0.452	0.671	338.9	0.227	24.80	0.678	0.816	228.1	0.162	33.90	0.952	0.970	78.0	0.065
RP-TC$_{\text{mcp}}$	**30.73**	**0.893**	**0.941**	**110.4**	**0.086**	**36.61**	**0.964**	**0.980**	**57.0**	**0.050**	**49.52**	**0.997**	**0.998**	**14.8**	**0.015**
RP-TC$_{\text{scad}}$	**30.60**	**0.890**	**0.938**	**112.1**	**0.086**	**36.30**	**0.962**	**0.979**	**58.9**	**0.052**	**49.41**	**0.997**	**0.998**	**15.0**	**0.015**

Table 4. The performance comparison of 7 competing TC methods with different sampling rates on *jellybeans* image.

Method	10%					20%					50%				
	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM
MC-ALM [3]	19.71	0.443	0.731	473.4	0.490	22.95	0.647	0.821	326.2	0.383	30.87	0.914	0.950	131.4	0.195
HaLRTC [4]	19.85	0.521	0.758	464.9	0.416	24.59	0.753	0.872	269.5	0.298	33.99	0.958	0.975	91.4	0.134
t-SVD [9]	27.53	0.784	0.890	191.6	0.360	32.34	0.901	0.947	114.5	0.258	41.87	0.982	0.990	42.9	0.107
TMac [8]	25.24	0.623	0.806	250.5	0.476	28.13	0.755	0.864	180.3	0.377	35.54	0.942	0.963	77.3	0.168
RP-TC$_{\text{trace}}$	19.77	0.545	0.770	468.8	0.434	25.75	0.804	0.898	234.5	0.296	38.08	0.982	0.989	56.6	0.095
RP-TC$_{\text{mcp}}$	**31.02**	**0.894**	**0.940**	**127.2**	**0.228**	**37.24**	**0.972**	**0.983**	**62.9**	**0.120**	**48.85**	**0.997**	**0.998**	**17.9**	**0.032**
RP-TC$_{\text{scad}}$	**31.07**	**0.896**	**0.940**	**126.5**	**0.222**	**36.90**	**0.970**	**0.981**	**65.3**	**0.122**	**48.79**	**0.997**	**0.998**	**18.0**	**0.032**

Table 5. The performance comparison of 7 competing TC methods with different sampling rates on *peppers* image.

Method	10%					20%					50%				
	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM
MC-ALM [3]	32.35	0.909	0.923	126.3	0.144	36.74	0.959	0.963	76.2	0.090	43.52	0.991	0.993	36.4	0.039
HaLRTC [4]	35.38	0.950	0.956	87.4	0.103	41.55	0.984	0.985	43.4	0.057	52.63	0.998	0.998	12.2	0.022
t-SVD [9]	40.30	0.962	0.962	53.7	0.121	46.78	0.989	0.987	25.6	0.061	56.70	0.998	0.998	8.7	0.023
TMac [8]	**43.22**	**0.983**	**0.981**	**35.9**	0.073	**48.62**	0.994	0.993	19.2	0.048	54.70	0.998	0.998	9.7	0.024
RP-TC$_{\text{trace}}$	32.26	0.921	0.937	125.2	0.136	41.92	0.986	0.986	41.4	0.057	53.55	0.999	0.999	10.9	0.021
RP-TC$_{\text{mcp}}$	**40.64**	**0.978**	**0.979**	**48.3**	**0.069**	**48.25**	**0.996**	**0.995**	**20.2**	**0.036**	**58.21**	**0.999**	**0.999**	**6.5**	**0.016**
RP-TC$_{\text{scad}}$	40.52	**0.978**	**0.979**	48.9	**0.071**	47.97	**0.995**	**0.995**	**20.8**	**0.037**	**57.89**	**0.999**	**0.999**	**6.7**	**0.016**

Table 6. The performance comparison of 7 competing TC methods with different sampling rates on *watercolors* image.

Method	10%					20%					50%				
	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM
MC-ALM [3]	22.73	0.608	0.795	197.2	0.150	25.30	0.750	0.858	146.7	0.111	31.23	0.924	0.950	74.2	0.060
HaLRTC [4]	23.56	0.699	0.808	178.8	0.127	27.41	0.843	0.895	115.0	0.084	36.34	0.974	0.981	41.2	0.041
t-SVD [9]	29.93	0.882	0.931	88.6	0.103	35.27	0.955	0.972	49.8	0.067	44.41	0.991	0.994	20.0	0.032
TMac [8]	27.09	0.783	0.875	119.3	0.119	30.05	0.876	0.919	84.8	0.092	38.20	0.974	0.982	33.2	0.045
RP-TC$_{\text{trace}}$	23.12	0.687	0.803	188.2	0.145	27.94	0.860	0.905	108.1	0.086	38.30	0.982	0.987	32.8	0.037
RP-TC$_{\text{mcp}}$	**31.84**	**0.929**	**0.950**	**68.6**	**0.074**	**37.75**	**0.977**	**0.984**	**35.0**	**0.045**	**48.40**	**0.996**	**0.998**	**11.0**	**0.024**
RP-TC$_{\text{scad}}$	**30.91**	**0.913**	**0.938**	**76.5**	**0.075**	**36.88**	**0.972**	**0.980**	**38.5**	**0.047**	**48.38**	**0.996**	**0.998**	**11.0**	**0.024**

Table 7. The performance comparison of 7 competing TC methods with different sampling rates on *Scene 4* of Natural Scenes 2002.

Method	10%					20%					50%				
	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM
MC-ALM [3]	33.94	0.842	0.923	127.9	0.113	36.75	0.901	0.958	92.1	0.088	42.89	0.972	0.992	44.7	0.047
HaLRTC [4]	34.87	0.883	0.918	116.3	0.085	38.45	0.933	0.965	77.3	0.065	46.10	0.986	0.996	31.7	0.032
t-SVD [9]	36.77	0.901	0.963	92.6	0.103	40.36	0.948	0.983	62.0	0.074	48.44	0.990	0.998	25.4	0.032
TMac [8]	36.50	0.880	0.965	92.8	0.110	40.57	0.951	0.985	58.1	0.068	45.51	0.983	0.996	32.4	0.035
RP-TC$_{\text{trace}}$	34.58	0.883	0.920	120.9	0.092	39.14	0.942	0.971	72.2	0.063	48.05	0.991	0.998	26.0	0.027
RP-TC$_{\text{mcp}}$	**40.21**	**0.946**	**0.974**	**64.2**	**0.055**	**46.04**	**0.984**	**0.995**	**33.2**	**0.034**	**54.31**	**0.998**	**0.999**	**12.7**	**0.014**
RP-TC$_{\text{scad}}$	**39.48**	**0.937**	**0.968**	**70.1**	**0.058**	**45.90**	**0.983**	**0.995**	**33.8**	**0.034**	**54.28**	**0.998**	**0.999**	**12.8**	**0.014**

Table 8. The performance comparison of 7 competing TC methods with different sampling rates on *Scene 8* of Natural Scenes 2002.

Method	10%					20%					50%				
	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM	PSNR	SSIM	FSIM	ERGAS	SAM
MC-ALM [3]	25.93	0.751	0.865	258.6	0.149	29.44	0.854	0.931	171.9	0.112	37.08	0.967	0.989	71.1	0.055
HaLRTC [4]	26.98	0.823	0.870	229.8	0.101	31.63	0.913	0.951	133.9	0.072	40.80	0.985	0.995	46.4	0.034
t-SVD [9]	34.62	0.936	0.976	96.9	0.083	39.42	0.972	0.991	57.8	0.055	47.89	0.994	0.998	22.7	0.025
TMac [8]	26.59	0.846	0.941	137.0	0.140	33.06	0.939	0.977	81.2	0.082	44.23	0.979	0.994	44.6	0.041
RP-TC$_{\text{trace}}$	26.59	0.826	0.872	240.7	0.111	33.06	0.935	0.966	113.6	0.070	44.22	0.993	0.998	31.3	0.026
RP-TC$_{\text{mcp}}$	**38.54**	**0.970**	**0.990**	**61.0**	**0.053**	**43.73**	**0.990**	**0.997**	**33.4**	**0.029**	**51.20**	**0.998**	**0.999**	**14.3**	**0.015**
RP-TC$_{\text{scad}}$	**37.08**	**0.961**	**0.984**	**71.4**	**0.055**	**43.31**	**0.989**	**0.997**	**35.0**	**0.030**	**51.20**	**0.998**	**0.999**	**14.4**	**0.015**

Table 9. The performance comparison of 7 competing TRPCA methods on *balloons* image.

	PSNR	SSIM	FSIM	ERGAS	SAM
Noisy image	13.05	0.057	0.282	939.3	0.908
TDL [5]	26.65	0.749	0.885	197.9	0.232
RPCA [3]	28.23	0.548	0.726	163.1	0.410
HoRPCA [1]	31.03	**0.929**	**0.950**	118.4	**0.197**
t-SVD [9]	26.60	0.715	0.879	198.1	0.249
RP-TRPCA$_{\text{trace}}$	31.24	**0.929**	**0.949**	115.6	**0.197**
RP-TRPCA$_{\text{mcp}}$	**34.19**	0.837	0.895	**82.8**	0.211
RP-TRPCA$_{\text{scad}}$	**33.89**	0.808	0.861	**95.7**	0.227

Table 10. The performance comparison of 7 competing TRPCA methods on *beads* image.

	PSNR	SSIM	FSIM	ERGAS	SAM
Noisy image	13.69	0.175	0.515	554.6	0.622
TDL [5]	26.98	0.803	**0.904**	120.3	0.154
RPCA [3]	27.29	0.699	0.870	115.9	0.277
HoRPCA [1]	26.36	0.806	0.855	129.0	**0.118**
t-SVD [9]	**29.75**	**0.823**	0.894	**90.4**	**0.133**
RP-TRPCA$_{\text{trace}}$	26.44	0.807	0.857	127.9	0.162
RP-TRPCA$_{\text{mcp}}$	**30.48**	**0.840**	**0.922**	**82.1**	0.150
RP-TRPCA$_{\text{scad}}$	29.28	0.757	0.895	92.0	0.185

Table 11. The performance comparison of 7 competing TRPCA methods on *chart and stuffed toy* image.

	PSNR	SSIM	FSIM	ERGAS	SAM
Noisy image	12.86	0.121	0.121	929.4	1.021
TDL [5]	26.03	0.594	0.890	202.7	0.274
RPCA [3]	29.43	0.714	0.837	137.8	0.586
HoRPCA [1]	28.83	**0.871**	0.902	148.0	0.234
t-SVD [9]	30.43	0.690	0.855	122.4	0.450
RP-TRPCA$_{\text{trace}}$	28.97	**0.873**	0.904	145.6	0.234
RP-TRPCA$_{\text{mcp}}$	**33.06**	0.859	**0.941**	**90.6**	**0.221**
RP-TRPCA$_{\text{scad}}$	**32.86**	0.857	**0.939**	**92.7**	**0.222**

Table 12. The performance comparison of 7 competing TRPCA methods on *cloth* image.

	PSNR	SSIM	FSIM	ERGAS	SAM
Noisy image	13.45	0.199	0.604	872.3	0.671
TDL [5]	24.90	0.700	0.868	229.7	0.184
RPCA [3]	26.60	0.737	**0.888**	178.5	0.195
HoRPCA [1]	24.62	0.624	0.736	230.2	0.140
t-SVD [9]	28.24	**0.792**	**0.893**	150.3	0.132
RP-TRPCA$_{\text{trace}}$	24.74	0.632	0.743	226.9	0.139
RP-TRPCA$_{\text{mcp}}$	**28.29**	**0.791**	0.878	**149.5**	**0.114**
RP-TRPCA$_{\text{scad}}$	**28.33**	0.786	0.872	**149.3**	**0.110**

Table 13. The performance comparison of 7 competing TRPCA methods on *egyptian statue* image.

	PSNR	SSIM	FSIM	ERGAS	SAM
Noisy image	12.55	0.040	0.236	1774.0	1.242
TDL [5]	25.91	0.357	0.842	380.4	0.441
RPCA [3]	31.48	0.608	0.781	199.0	0.804
HoRPCA [1]	33.44	**0.896**	0.932	163.5	0.417
t-SVD [9]	32.49	0.594	0.798	177.3	0.709
RP-TRPCA$_{\text{trace}}$	33.54	**0.901**	0.933	161.7	0.416
RP-TRPCA$_{\text{mcp}}$	**35.90**	0.777	**0.940**	**122.3**	**0.412**
RP-TRPCA$_{\text{scad}}$	**35.72**	0.771	**0.938**	**124.9**	**0.412**

Table 14. The performance comparison of 7 competing TRPCA methods on *feathers* image.

	PSNR	SSIM	FSIM	ERGAS	SAM
Noisy image	12.88	0.099	0.382	1061.2	1.005
TDL [5]	26.20	0.642	0.895	227.2	0.298
RPCA [3]	29.00	0.658	0.828	165.5	0.517
HoRPCA [1]	29.07	0.871	0.897	164.3	0.232
t-SVD [9]	30.50	0.695	0.851	138.8	0.370
RP-TRPCA$_{\text{trace}}$	29.21	0.874	0.899	161.6	0.232
RP-TRPCA$_{\text{mcp}}$	**33.19**	**0.877**	**0.927**	**102.2**	**0.214**
RP-TRPCA$_{\text{scad}}$	**33.12**	**0.876**	**0.926**	**103.0**	**0.214**

Table 15. The performance comparison of 7 competing TRPCA methods on *flowers* image.

	PSNR	SSIM	FSIM	ERGAS	SAM
Noisy image	12.70	0.067	0.333	1417.6	1.121
TDL [5]	25.94	0.476	0.881	306.0	0.439
RPCA [3]	30.17	0.633	0.826	182.5	0.680
HoRPCA [1]	30.63	**0.823**	0.880	176.6	0.389
t-SVD [9]	31.20	0.624	0.842	162.2	0.568
RP-TRPCA$_{\text{trace}}$	30.74	**0.825**	0.882	174.3	0.386
RP-TRPCA$_{\text{mcp}}$	**33.84**	0.766	**0.917**	**120.7**	**0.368**
RP-TRPCA$_{\text{scad}}$	**33.74**	0.764	**0.916**	**122.0**	**0.368**

Table 16. The performance comparison of 7 competing TRPCA methods on *photo and face* image.

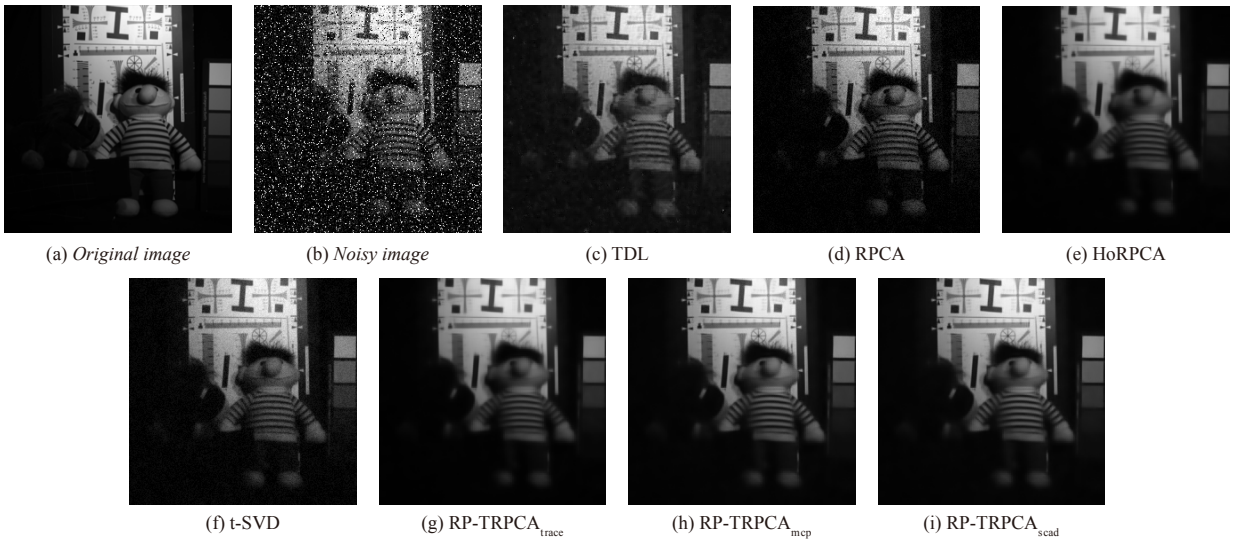
	PSNR	SSIM	FSIM	ERGAS	SAM
Noisy image	12.71	0.089	0.363	1320.5	1.137
TDL [5]	26.03	0.522	0.894	284.2	0.308
RPCA [3]	30.20	0.715	0.851	175.5	0.682
HoRPCA [1]	30.80	**0.884**	0.916	163.7	0.291
t-SVD [9]	31.53	0.703	0.869	150.9	0.566
RP-TRPCA$_{\text{trace}}$	30.92	**0.890**	0.917	161.5	0.291
RP-TRPCA$_{\text{mcp}}$	**34.38**	0.840	**0.947**	**108.8**	**0.282**
RP-TRPCA$_{\text{scad}}$	**34.20**	0.837	**0.945**	**111.0**	**0.283**

Table 17. The performance comparison of 7 competing TRPCA methods on *pompoms* image.

	PSNR	SSIM	FSIM	ERGAS	SAM
Noisy image	13.22	0.106	0.444	876.8	0.798
TDL [5]	26.82	0.782	0.919	183.1	0.204
RPCA [3]	27.47	0.631	0.853	167.4	0.298
HoRPCA [1]	28.79	0.873	0.910	143.4	0.154
t-SVD [9]	30.12	0.730	0.898	123.6	0.198
RP-TRPCA$_{\text{trace}}$	28.85	0.874	0.910	142.4	0.153
RP-TRPCA$_{\text{mcp}}$	**33.00**	**0.899**	**0.934**	**88.9**	**0.134**
RP-TRPCA$_{\text{scad}}$	**32.92**	**0.898**	**0.933**	**89.8**	**0.134**

Table 18. The performance comparison of 7 competing methods on *watercolors* image.

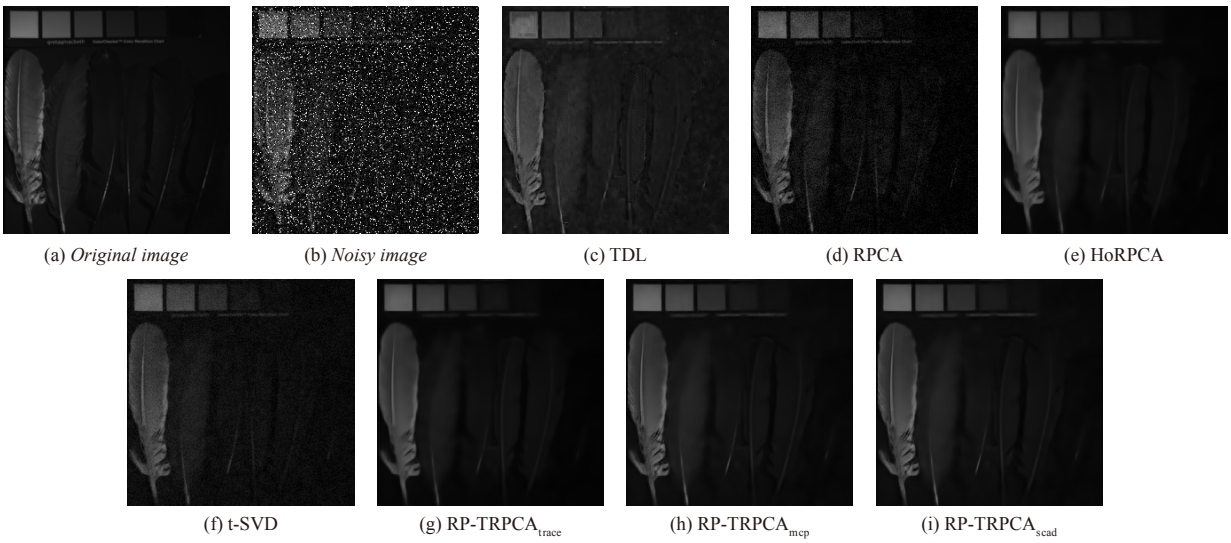
	PSNR	SSIM	FSIM	ERGAS	SAM
Noisy image	12.95	0.209	0.588	1103.2	0.937
TDL [5]	24.79	0.683	0.893	282.0	0.333
RPCA [3]	26.71	0.754	0.896	222.1	0.460
HoRPCA [1]	24.10	0.712	0.819	302.6	0.296
t-SVD [9]	27.20	0.765	0.896	215.2	0.322
RP-TRPCA$_{\text{trace}}$	23.94	0.712	0.819	307.6	0.298
RP-TRPCA$_{\text{mcp}}$	**28.66**	**0.829**	**0.914**	**181.1**	**0.259**
RP-TRPCA$_{\text{scad}}$	**28.64**	**0.826**	**0.913**	**182.1**	**0.263**



(a) *Original image* (b) *Noisy image* (c) TDL (d) RPCA (e) HoRPCA

(f) t-SVD (g) RP-TRPCA$_{\text{trace}}$ (h) RP-TRPCA$_{\text{mcp}}$ (i) RP-TRPCA$_{\text{scad}}$

Figure 1. (a) The original image from *chart and stuffed toy*; (b) The corresponding noisy image; (c)-(i) The recovered images by TDL[5], RPCA [3], HoRPCA [1], t-SVD [9], RP-TRPCA$_{\text{trace}}$, RP-TRPCA$_{\text{mcp}}$ and RP-TRPCA$_{\text{scad}}$, respectively.



(a) *Original image* (b) *Noisy image* (c) TDL (d) RPCA (e) HoRPCA

(f) t-SVD (g) RP-TRPCA$_{\text{trace}}$ (h) RP-TRPCA$_{\text{mcp}}$ (i) RP-TRPCA$_{\text{scad}}$

Figure 2. (a) The original image from *feathers*; (b) The corresponding noisy image; (c)-(i) The recovered images by TDL[5], RPCA [3], HoRPCA [1], t-SVD [9], RP-TRPCA$_{\text{trace}}$, RP-TRPCA$_{\text{mcp}}$ and RP-TRPCA$_{\text{scad}}$, respectively.

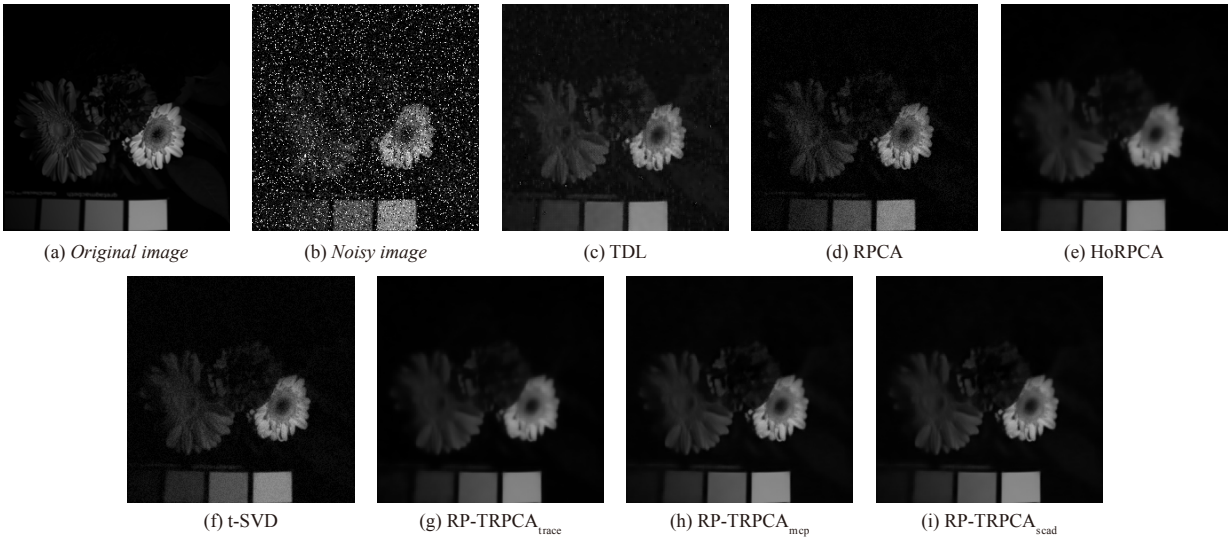


Figure 3. (a) The original image from *flowers*; (b) The corresponding noisy image; (c)-(i) The recovered images by TDL[5], RPCA [3], HoRPCA [1], t-SVD [9], RP-TRPCA$_{trace}$, RP-TRPCA$_{mcp}$ and RP-TRPCA$_{scad}$, respectively.

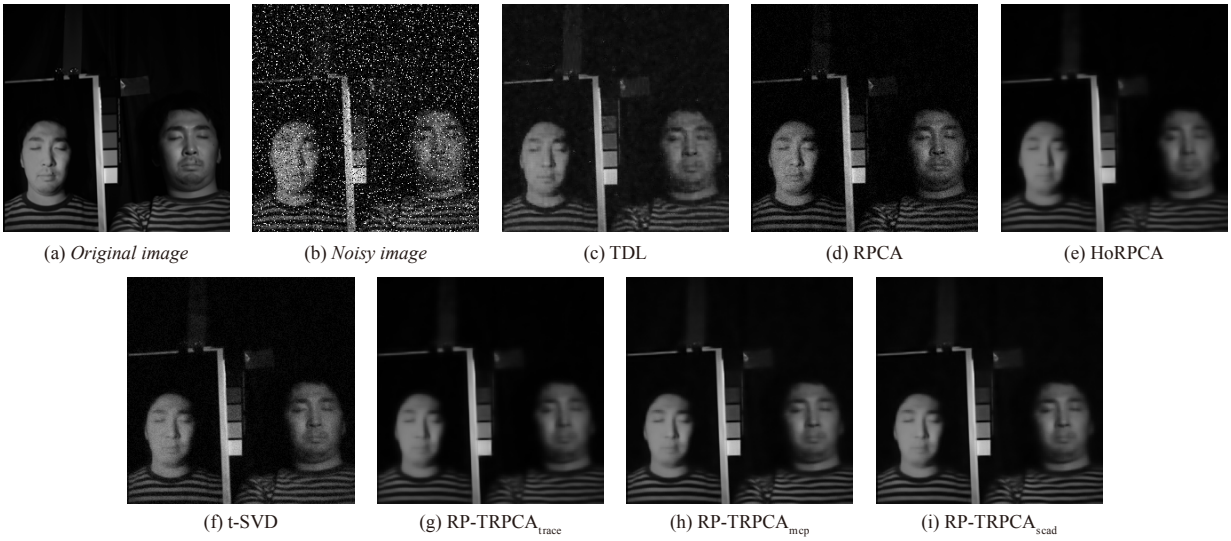


Figure 4. (a) The original image from *photo and face*; (b) The corresponding noisy image; (c)-(i) The recovered images by TDL[5], RPCA [3], HoRPCA [1], t-SVD [9], RP-TRPCA$_{trace}$, RP-TRPCA$_{mcp}$ and RP-TRPCA$_{scad}$, respectively.