

T-SVD Based Non-convex Tensor Completion and Robust Principal Component Analysis

Tao Li

Department of Information Science
 School of Mathematical Sciences and LMAM
 Peking University, Beijing, China
 Email: li_tao@pku.edu.cn

Jinwen Ma

Department of Information Science
 School of Mathematical Sciences and LMAM
 Peking University, Beijing, China
 Email: jwma@math.pku.edu.cn

Abstract—Tensor completion and robust principal component analysis have been widely used in machine learning while the key problem relies on the minimization of a tensor rank that is very challenging. A common way to tackle this difficulty is to approximate the tensor rank with ℓ_1 -norm of the singular values solved by the Tensor Singular Value Decomposition (T-SVD). Besides, the sparsity of a tensor is also measured with ℓ_1 -norm. However, the ℓ_1 penalty is essentially biased and thus the result will deviate. In order to sidestep the bias, we propose a novel non-convex tensor rank surrogate function and a novel non-convex sparsity measure. In this new setting by using the concavity instead of the convexity, a majorization minimization algorithm is further designed for tensor completion and robust principal component analysis. Furthermore, we analyze its theoretical properties. Finally, the experiments on both natural and hyperspectral images demonstrate the efficacy and efficiency of our proposed method.

I. INTRODUCTION

Tensors [1], or multi-way arrays, have been extensively used in computer vision [2], [3], signal processing and machine learning [4], [5]. Due to technical reasons, tensors in most applications are incomplete or polluted. Generically, recovering a tensor from corrupted observations is an inverse problem, which is ill-posed without prior knowledge. However, in real applications, entries in a tensor are usually highly correlated, which means a high-dimensional tensor is intrinsically determined by low-dimensional factors. Exploiting such low-dimensional structure makes it possible to restore tensors from limited or corrupted observations. Mathematically, this prior knowledge is equivalent to assume the tensors are low-rank.

In this work, we mainly consider two tensor recovery problems: tensor completion and tensor robust principal component analysis (TRPCA). The tensor completion problem is to estimate the missing values in tensors from partially observed data, while TRPCA aims to decompose a tensor into a low-rank tensor and sparse tensor. In the case of 2-order tensor, *i.e.*, the matrix case, both problems have been investigated thoroughly [6], [7], [8], [9]. Since the concept of a tensor is an extension of the matrix, it is natural to employ matrix recovery methods to tensors. Most matrix recovery methods are optimization-based, penalizing rank surrogate function, or/and certain sparsity measure. Similar methods have been developed for tensors. Tensor ℓ_1 -norm is often used as sparsity measure. However, the concept of tensor rank is far more

complicated than matrix rank, thus there are various surrogate functions for tensor rank, such as the sum of the nuclear norm (SNN) [2], tensor nuclear norm (TNN) [10] and twisted tensor nuclear norm (t-TNN) [11].

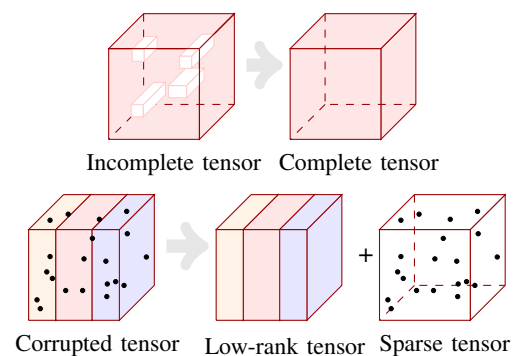


Fig. 1. An illustration of tensor completion and TRPCA.

As in the matrix case, the choices of rank surrogate function and sparsity measure substantially influence the final results. The nuclear norm of a matrix is equivalent to the ℓ_1 -norm of its singular value. However, as indicated by Fan and Li [12], ℓ_1 -norm over-penalizes large entries of vectors. Smoothly clipped absolute deviation (SCAD) penalty [12] and minimax concave plus (MCP) penalty [13] were proposed as ideal penalty functions, and their superiority over ℓ_1 -norm has been demonstrated in [13], [12], [14], [15]. This fact inspires us that nuclear norm based tensor rank surrogate functions and ℓ_1 -norm based tensor sparsity measure may suffer from a similar problem. To alleviate such phenomena, we propose to use non-convex penalties (SCAD and MCP) instead of an ℓ_1 -norm in TNN and tensor sparsity measures.

However, the introduction of non-convex penalties makes optimization problems even harder to solve. For example, TNN based TRPCA [10] is a convex optimization model, thus can be efficiently solved by alternating direction multiplier method (ADMM) [16]. Once we replace ℓ_1 -norm by SCAD or MCP, the problem is not convex anymore. Therefore, we apply the majorization minimization algorithm [17], [18] to solve the non-convex optimization problems and analyze the theoretical properties of these algorithms. Based on the proposed non-convex tensor completion and TRPCA models and their corresponding MM algorithms, we conduct experiments on natural

images and multispectral images to validate the efficacy of the proposed methods.

II. RELATED WORK

Tensor recovery. For tensor completion, one seminal work is [2], in which SNN was proposed and three different algorithms for solving SNN based TC were devised. Zhao *et al.* proposed Bayesian CANDECOMP/PARAFAC (CP) tensor factorization model in [3]. The highlights of [3] include automatic rank determination property, full Bayesian treatment, and uncertainty quantification. Kilmer and Martin proposed a new tensor singular value decomposition (t-SVD) based on discrete Fourier transform for 3–order tensors in [19], [20]. The key point is that t-SVD offers an efficient way to define tensor nuclear norm (TNN), which has been extensively used in tensor recovery recently [21], [10], [22], [23]. Furthermore, Lu *et al.* proved the exact recovery property of their proposed TRPCA model under certain suitable assumptions [10], [23].

Non-convex penalties. Wang and Zhang [24] developed a non-convex optimization model for the low-rank matrix recovery problem. Cao *et al.* [25] applied folded-concave penalties in SNN, while Ji *et al.* [26] used log determinant penalty instead. Zhao *et al.* [27] proposed to use the product of nuclear norm instead of the sum of the nuclear norm, which has a natural physical meaning. Besides, they also considered non-convex penalties such as SCAD and MCP. One major difference between our work and [25], [27] is that our methods are based on t-SVD, while [25], [27] transform a tensor to matrices simply via unfolding. Recently, Jiang *et al.* [28] and Xu *et al.* [29] introduced non-convex penalties to TNN, but neither MCP nor SCAD was considered. Besides, our work not only improves the tensor rank surrogate function but also modifies the tensor sparsity measure. Yokota *et al.* [30] proposed a Tucker decomposition based non-convex tensor completion model for a case where all of the elements in some continuous slices are missing. Yao *et al.* [31] also considered non-convex regularizers for tensor recovery and devised an efficient solver based on the proximal average algorithm. However, the non-convex penalties in [31] are based on SNN, while ours are based on TNN.

III. NOTATIONS AND PRELIMINARIES

A. Notations

Throughout this paper, we use calligraphic letters to denote 3-way tensors, *e.g.*, $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$. The (i, j, k) -th element of \mathcal{A} may be denoted by $\mathcal{A}(i, j, k)$ or \mathcal{A}_{ijk} alternatively. The k -th frontal slice of \mathcal{A} is defined as $\mathcal{A}(:, :, k)$, which is an $n_1 \times n_2$ matrix. For brevity, we use $A^{(k)}$ to denote $\mathcal{A}(:, :, k)$. The (i, j) -th tube of \mathcal{A} is defined as $\mathcal{A}(i, j, :)$, which is a vector of length n_3 . The inner product of two 3-way tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ is defined as $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_k \text{Tr}((A^{(k)})^* B^{(k)})$. We use $|\mathcal{A}|$ to denote the tensor with (i, j, k) -th element equals to $|\mathcal{A}_{ijk}|$. Similar to vectors and matrices, we can also define various norms of tensors. We denote ℓ_1 -norm by $\|\mathcal{A}\|_1 = \sum_{ijk} |\mathcal{A}_{ijk}|$, ℓ_∞ -norm by

$\|\mathcal{A}\|_\infty = \max_{ijk} |\mathcal{A}_{ijk}|$ and Frobenius norm by $\|\mathcal{A}\|_F = \sqrt{\sum_{ijk} |\mathcal{A}_{ijk}|^2}$. An $n_1 \times n_2 \times n_3$ tensor \mathcal{A} can be transformed to an $(n_1 n_3) \times (n_2 n_3)$ block diagonal matrix A whose blocks are the frontal slices $A^{(1)}, \dots, A^{(n_3)}$. Both this transform and its inverse are denoted by bdiag , and the meaning can be understood according to the type of the input.

Discrete Fourier transform (DFT) and inverse discrete Fourier transform (IDFT) are essential to the definitions in Section III-B. We use fft and ifft to denote DFT and IDFT applying to each tube of a 3-way tensor. We define $\bar{\mathcal{A}} = \text{fft}(\mathcal{A}, 3)$, and it is obvious that $\mathcal{A} = \text{ifft}(\bar{\mathcal{A}}, 3)$. Furthermore, we use $\bar{A} = \text{bdiag}(\bar{\mathcal{A}})$ to denote the block diagonal matrix whose blocks are frontal slices of $\bar{\mathcal{A}}$. With a little abuse of terminology, we say \mathcal{A} is in original domain and $\bar{\mathcal{A}}$ (or equivalently \bar{A}) is in transformation domain or Fourier domain.

B. T-Product and T-SVD

Definition 3.1 (T-product): [19], [23] Suppose $\mathcal{A} \in \mathbb{C}^{n_1 \times m \times n_3}$ and $\mathcal{B} \in \mathbb{C}^{m \times n_2 \times n_3}$, then the t-product $\mathcal{A} * \mathcal{B} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ is defined as

$$\mathcal{A} * \mathcal{B} = \text{ifft}(\text{bdiag}(\bar{A}\bar{B}), 3) \quad (1)$$

Note that Definition 3.1 is different from [19], [23] in form, but it is equivalent to the standard definitions. The reason why we choose this form is to avoid some cumbersome notations and better reveal the relationship between original domain and transformation domain. We may regard t-product as transforming the tensors by DFT, then multiplying corresponding frontal slices in Fourier domain, and finally transforming the result back to original domain by IDFT. It has been proved in [19], [23] that $\mathcal{C} = \mathcal{A} * \mathcal{B}$ is equivalent to $\bar{\mathcal{C}} = \bar{A}\bar{B}$.

Before we introduce T-SVD, we need some further definitions, which are direct extensions of the corresponding definitions in the matrix case.

Definition 3.2 (Conjugate transpose): [19], [23] Suppose $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, the conjugate transpose of \mathcal{A} is denoted by $\mathcal{A}^* \in \mathbb{C}^{n_2 \times n_1 \times n_3}$ whose first frontal slice equals to $(A^{(1)})^*$ and whose k -th frontal slice ($k = 2, 3, \dots, n_3$) equals to $(A^{(n_3+2-k)})^*$.

Definition 3.3 (Identity tensor): [19], [23] The identity tensor $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$ is the tensor whose first frontal slice is the $n \times n$ identity matrix and whose other slices are zeros.

Definition 3.4 (Orthogonal tensor): [19], [23] A tensor $\mathcal{Q} \in \mathbb{R}^{n \times n \times n_3}$ is orthogonal if $\mathcal{Q} * \mathcal{Q}^* = \mathcal{Q}^* * \mathcal{Q} = \mathcal{I}$.

Definition 3.5 (F-diagonal): [19], [23] A tensor is said to be f-diagonal if its every frontal slice is a diagonal matrix.

Theorem 3.1 (T-SVD): [19], [23] Suppose $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Then there exists tensors $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ and $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ such that $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$. Furthermore, \mathcal{U} and \mathcal{V} are orthogonal, while \mathcal{S} is f-diagonal.

Note that $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$ in original domain is equivalent to $\bar{\mathcal{A}} = \bar{\mathcal{U}}\bar{\mathcal{S}}\bar{\mathcal{V}}^*$ in Fourier domain. Intuitively, we can obtain the T-SVD of \mathcal{A} by calculating SVD of each frontal slice $\bar{A}^{(k)}$ in frequency domain, *i.e.*, $\bar{A}^{(k)} = \bar{U}^{(k)}\bar{S}^{(k)}(\bar{V}^{(k)})^*$, then

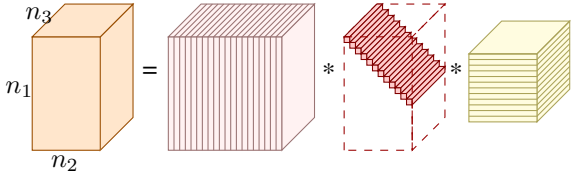


Fig. 2. An illustration of the t-SVD of an $n_1 \times n_2 \times n_3$ tensor.

transforming $\bar{U}, \bar{S}, \bar{V}$ to original domain by IDFT. However, as indicated in [23], this method may result in complex entries due to non-uniqueness of matrix SVD. We omit the detailed algorithm for calculating T-SVD due to space limit, and refer to [23] for further discussions.

The concept of rank for tensors is very complicated. There are various definitions of tensor rank [1], [4], [5], and most of them are NP-complete. The rank of a matrix is equivalent to the number of its non-zero singular values, and we often use nuclear norm (the sum of all singular values) as a surrogate function for matrix rank. Intuitively, we may extend the concept of the nuclear norm to the tensor case, and the extension may be a reasonable surrogate for tensor rank.

Definition 3.6 (Tensor nuclear norm): [19], [23] Let $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$ be the t-SVD of \mathcal{A} , the nuclear norm of \mathcal{A} is defined as $\|\mathcal{A}\|_* = \sum_i \mathcal{S}(i, i, 1)$.

It has been proved in [23] that Definition 3.6 is the convex envelope of tensor average rank. Besides, the tensor nuclear norm is the dual norm of the tensor spectral norm, which is consistent with the matrix case. At first glance, the definition above may be a little amazing since only the first frontal slice of \mathcal{S} is used. According to the definition of IDFT, we have $\mathcal{S}(i, i, 1) = \frac{1}{n_3} \sum_k \bar{\mathcal{S}}(i, i, k)$. Thus, in the transformation domain, the tensor nuclear norm is equal to the sum of all singular values of all frontal slices up to a constant factor.

C. Non-convex Penalties: SCAD and MCP

As indicated in [12], an ideal penalty function should result in an estimator with three properties: unbiasedness, sparsity and continuity. Smoothly clipped absolute deviation (SCAD) was proposed in [12] to improve the properties of the ℓ_1 penalty, which does not satisfies the three properties simultaneously.

Definition 3.7 (SCAD): [12] For some $\gamma > 2$ and $\lambda > 0$, the SCAD function is given by

$$\varphi_{\lambda, \gamma}^{\text{SCAD}}(t) = \begin{cases} \lambda|t| & \text{if } |t| \leq \lambda, \\ \frac{\gamma\lambda|t| - 0.5(t^2 + \lambda^2)}{\gamma - 1} & \text{if } \lambda < |t| < \gamma\lambda, \\ \frac{\gamma + 1}{2} \lambda^2 & \text{if } |t| > \gamma\lambda. \end{cases} \quad (2)$$

A continuous, nearly unbiased and accurate variable selection penalty called minimax concave penalty (MCP) was proposed in [13]. The precise definition is given as follows.

Definition 3.8 (MCP): [13] For some $\gamma > 1$ and $\lambda > 0$, the MCP function is given by

$$\varphi_{\lambda, \gamma}^{\text{MCP}}(t) = \begin{cases} \lambda|t| - \frac{t^2}{2\gamma} & \text{if } |t| < \gamma\lambda, \\ \frac{\gamma\lambda^2}{2} & \text{if } |t| \geq \gamma\lambda. \end{cases} \quad (3)$$

It is well known that ℓ_1 -norm penalty over-penalizes large components. However, in SCAD and MCP, the penalty remains constant once the variable is larger than a threshold. Besides, we point out that as $\gamma \rightarrow \infty$, we have $\varphi_{\lambda, \gamma}^{\text{SCAD}}(t) \rightarrow \lambda|t|$ and $\varphi_{\lambda, \gamma}^{\text{MCP}}(t) \rightarrow \lambda|t|$ pointwisely. Last but not least, if we restrict $t \geq 0$, or equivalently view SCAD and MCP as functions of $|t|$, then they are concave functions. In the following, we use $\varphi_{\lambda, \gamma}(t)$ to denote SCAD or MCP alternatively.

The effects of λ and γ in SCAD and MCP can be understood intuitively by considering $\varphi_{\lambda, \gamma}(t) \rightarrow \lambda|t|$. Roughly, λ controls the relative importance of the penalty, and γ controls how similar is $\varphi_{\lambda, \gamma}(t)$ compared with $\lambda|t|$.

IV. THEORETICAL FOUNDATIONS

A. A Novel Tensor Sparsity Measure

The ℓ_1 -norm has been widely used as a sparsity measure in statistics, machine learning and computer vision. For tensors, the tensor ℓ_1 -norm plays a vital role in TRPCA [27], [10], [23]. However, ℓ_1 -norm penalty over-penalizes larger entries and may result in biased estimator. Therefore, we propose to use SCAD or MCP instead of the ℓ_1 -norm penalty. The novel tensor sparsity measure is defined as

$$\Phi_{\lambda, \gamma}(\mathcal{A}) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \varphi_{\lambda, \gamma}(\mathcal{A}_{ijk}). \quad (4)$$

Here, we may set $\varphi_{\lambda, \gamma}$ to be $\varphi_{\lambda, \gamma}^{\text{SCAD}}$ or $\varphi_{\lambda, \gamma}^{\text{MCP}}$. We have the following properties.

Proposition 4.1: For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\Phi_{\lambda, \gamma}(\mathcal{A})$ satisfies:

- (i) $\Phi_{\lambda, \gamma}(\mathcal{A}) \geq 0$ with the equality holds iff $\mathcal{A} = 0$;
- (ii) $\Phi_{\lambda, \gamma}(\mathcal{A})$ is concave with respect to $|\mathcal{A}|$;
- (iii) $\Phi_{\lambda, \gamma}(\mathcal{A})$ is increasing in γ , $\Phi_{\lambda, \gamma}(\mathcal{A}) \leq \lambda\|\mathcal{A}\|_1$ and $\lim_{\gamma \rightarrow \infty} \Phi_{\lambda, \gamma}(\mathcal{A}) = \lambda\|\mathcal{A}\|_1$.

B. A Novel Tensor Rank Penalty

In this part, we always assume $\lambda = 1$. Similar to tensor nuclear norm, we can apply SCAD or MCP to the singular values of a tensor. However, this may result in difficulty in optimization algorithms. Instead, we propose to apply penalty function to all singular values in Fourier domain. More precisely, suppose \mathcal{A} has t-SVD $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$, we define the γ -norm of \mathcal{A} as

$$\|\mathcal{A}\|_{\gamma} = \frac{1}{n_3} \sum_{i,k} \varphi_{1, \gamma}(\bar{\mathcal{S}}(i, i, k)). \quad (5)$$

The tensor γ -norm enjoys the following properties.

Proposition 4.2: For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, suppose \mathcal{A} has t-SVD $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$, then $\|\mathcal{A}\|_{\gamma}$ satisfies:

- (i) $\|\mathcal{A}\|_{\gamma} \geq 0$ with equality holds iff $\mathcal{A} = 0$;
- (ii) $\|\mathcal{A}\|_{\gamma}$ is increasing in γ , $\|\mathcal{A}\|_{\gamma} \leq \|\mathcal{A}\|_*$ and $\lim_{\gamma \rightarrow \infty} \|\mathcal{A}\|_{\gamma} = \|\mathcal{A}\|_*$;
- (iii) $\|\mathcal{A}\|_{\gamma}$ is concave with respect to $\{\bar{\mathcal{S}}(i, i, k)\}_{i,k}$;
- (iv) $\|\mathcal{A}\|_{\gamma}$ is orthogonal invariant, *i.e.*, for any orthogonal tensors $\mathcal{P} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, $\mathcal{Q} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$, we have $\|\mathcal{P} * \mathcal{A} * \mathcal{Q}\|_{\gamma} = \|\mathcal{A}\|_{\gamma}$.

C. Generalized Thresholding Operators

We will use majorization minimization algorithm in Section V-A and V-B. In this part, we derive some properties that are vital to MM algorithm based on the concavity of SCAD and MCP. As mentioned in Section III-C, SCAP and MCP are continuous differentiable concave functions restricted on $[0, \infty)$, thus we can bound $\varphi_{\lambda, \gamma}(t)$ by its first-order Taylor expansion $\varphi_{\lambda, \gamma}(t_0) + \varphi'_{\lambda, \gamma}(t_0)(t - t_0)$. This observation leads to the following theorem.

Theorem 4.3: We can view $\Phi_{\lambda, \mu}(\mathcal{X})$ as a function of $|\mathcal{X}|$, and $\|\mathcal{X}\|_{\gamma}$ as a function of $\{\bar{\mathcal{S}}(i, j, k)\}_{i, j, k}$. For any \mathcal{X}^{old} , let

$$Q_{\lambda, \gamma}(\mathcal{X}|\mathcal{X}^{\text{old}}) = \Phi_{\lambda, \gamma}(\mathcal{X}^{\text{old}}) + \sum_{i, j, k} \varphi'_{\lambda, \gamma}(|\mathcal{X}_{ijk}^{\text{old}}|)(|\mathcal{X}_{ijk}| - |\mathcal{X}_{ijk}^{\text{old}}|),$$

$$Q_{\gamma}(\mathcal{X}|\mathcal{X}^{\text{old}}) = \|\mathcal{X}^{\text{old}}\|_{\gamma} + \frac{1}{n_3} \sum_{i, k} \varphi'_{1, \gamma}(\bar{\mathcal{S}}_{iik}^{\text{old}})(\bar{\mathcal{S}}_{iik} - \bar{\mathcal{S}}_{iik}^{\text{old}}), \quad (6)$$

then

$$Q_{\lambda, \gamma}(\mathcal{X}^{\text{old}}|\mathcal{X}^{\text{old}}) = \Phi_{\lambda, \gamma}(\mathcal{X}^{\text{old}}), \Phi_{\lambda, \gamma}(\mathcal{X}) \leq Q_{\lambda, \gamma}(\mathcal{X}|\mathcal{X}^{\text{old}}),$$

$$Q_{\gamma}(\mathcal{X}^{\text{old}}|\mathcal{X}^{\text{old}}) = \|\mathcal{X}^{\text{old}}\|_{\gamma}, \|\mathcal{X}\|_{\gamma} \leq Q_{\gamma}(\mathcal{X}|\mathcal{X}^{\text{old}}). \quad (7)$$

Due to the concavity of $\Phi_{\lambda, \gamma}(\mathcal{X})$ and $\|\mathcal{X}\|_{\gamma}$, optimization problems involving $\Phi_{\lambda, \gamma}(\mathcal{X})$ and $\|\mathcal{X}\|_{\gamma}$ are generally extremely difficult to solve. However, optimizing upper bounds given in Theorem 4.3 instead is relatively easy. It's well-known that soft thresholding operator $\mathcal{T}_{\lambda}(z) = \text{sgn}(z)[|z| - \lambda]_+$ is the proximal operator of ℓ_1 -norm. In the following, we introduce generalized thresholding operators based on \mathcal{T}_{λ} , then derive the proximal operators of $Q_{\lambda, \gamma}(\mathcal{X}|\mathcal{X}^{\text{old}})$ and $Q_{\gamma}(\mathcal{X}|\mathcal{X}^{\text{old}})$.

Definition 4.1 (Generalized soft thresholding): Suppose $\mathcal{X}, \mathcal{W} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the generalized soft thresholding operator is defined as

$$[\mathcal{T}_{\mathcal{W}}(\mathcal{X})]_{ijk} = \mathcal{T}_{\mathcal{W}_{ijk}}(\mathcal{X}_{ijk}). \quad (8)$$

Theorem 4.4: For $\forall \mu > 0$, let $\mathcal{W}_{ijk} = \varphi'_{\lambda, \gamma}(|\mathcal{X}_{ijk}^{\text{old}}|)/\mu$, then

$$\mathcal{T}_{\mathcal{W}}(\mathcal{Y}) = \arg \min_{\mathcal{X}} Q_{\lambda, \gamma}(\mathcal{X}|\mathcal{X}^{\text{old}}) + \frac{\mu}{2} \|\mathcal{X} - \mathcal{Y}\|_F^2. \quad (9)$$

Definition 4.2 (Generalized t-SVT): Suppose a 3-way tensor \mathcal{Y} has t-SVD $\mathcal{Y} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$, \mathcal{W} is a tensor with the same shape of \mathcal{Y} , the generalized tensor singular value thresholding operator is defined as

$$\mathcal{D}_{\mathcal{W}}(\mathcal{Y}) = \mathcal{U} * \tilde{\mathcal{S}} * \mathcal{V}^*, \quad (10)$$

where $\tilde{\mathcal{S}} = \text{ifft}(\mathcal{T}_{\mathcal{W}}(\mathcal{S}), 3)$.

Theorem 4.5: For $\forall \mu > 0$, let $\mathcal{W}_{ijk} = \delta_i^j \varphi'_{1, \gamma}(\bar{\mathcal{S}}_{iik}^{\text{old}})/\mu$ where δ_i^j is the Kronecker symbol, then

$$\mathcal{D}_{\mathcal{W}}(\mathcal{Y}) = \arg \min_{\mathcal{X}} Q_{\gamma}(\mathcal{X}|\mathcal{X}^{\text{old}}) + \frac{\mu}{2} \|\mathcal{X} - \mathcal{Y}\|_F^2. \quad (11)$$

V. PROPOSED NON-CONVEX TENSOR RECOVERY MODELS AND ALGORITHMS

A. Non-convex Tensor Completion

Given a partially observed tensor $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, tensor completion task aims to recover the full tensor \mathcal{X} which coincides with \mathcal{O} in the observed positions. Suppose the

Algorithm 1 MM algorithm for non-convex low-rank tensor completion

Input: Ω, \mathcal{O}

Hyper parameters: $\gamma, \mu_0, \rho, \mu_{\max}$

- 1: Initialize $\mathcal{X}^{\text{old}} = \mathcal{X}^0$ by $\mathcal{O} \otimes \Omega$ or other tensor completion algorithm.
- 2: **while** not converged **do**
- 3: Calculate $\bar{\mathcal{S}}^{\text{old}}$ and set $\mathcal{W}_{ijk}^t = \delta_i^j \varphi'_{1, \gamma}(\bar{\mathcal{S}}_{iik}^{\text{old}})/\mu$.
- 4: Initialize $\mathcal{X}_0^t = \mathcal{X}^t, \mathcal{Y}_0^t = 0$.
- 5: **while** not converged **do**
- 6: $\mathcal{M}_{l+1}^t = \mathcal{D}_{\mathcal{W}^t}(\mathcal{X}_l^t - \frac{1}{\mu_l} \mathcal{Y}_l^t)$
- 7: $\mathcal{X}_{l+1}^t = (\mathcal{M}_{l+1}^t + \frac{1}{\mu_l} \mathcal{Y}_l^t) \otimes (1 - \Omega) + \mathcal{O} \otimes \Omega$
- 8: $\mathcal{Y}_{l+1}^t = \mathcal{Y}_l^t + \mu_l (\mathcal{M}_{l+1}^t - \mathcal{X}_{l+1}^t)$
- 9: $\mu_{l+1} = \min(\rho \mu_l, \mu_{\max})$
- 10: **end while**
- 11: Update \mathcal{X}^{t+1} by the result of inner iteration
- 12: Set $\mathcal{X}^{\text{old}} = \mathcal{X}^{t+1}$
- 13: **end while**

observed positions are indexed by Ω , *i.e.*, $\Omega_{ijk} = 1$ denotes the (i, j, k) -th element is observed while $\Omega_{ijk} = 0$ denotes the (i, j, k) -th element is unknown. Based on low rank assumption, tensor completion can be modeled as

$$\min_{\mathcal{X}} \text{rank}(\mathcal{X}) \quad \text{s.t. } \mathcal{O}_{\Omega} = \mathcal{X}_{\Omega}. \quad (12)$$

Since the concept of rank is very complicated for tensors, many types of tensor rank or surrogate functions can be used in Equation 12. Here, we use the proposed tensor γ -norm,

$$\min_{\mathcal{X}} \|\mathcal{X}\|_{\gamma} \quad \text{s.t. } \mathcal{O}_{\Omega} = \mathcal{X}_{\Omega}. \quad (13)$$

We can set $\varphi_{1, \gamma}$ in Equation (13) to be SCAD or MCP. In the following we refer these non-convex tensor completion models as $\text{LRTC}_{\text{scad}}$ and LRTC_{mcp} respectively.

We apply majorization minimization algorithm to solve problem (13). Given \mathcal{X}^{old} , we minimize the upper bound of $\|\mathcal{X}\|_{\gamma}$ given in Theorem 4.3, which leads to

$$\min_{\mathcal{X}} Q_{\gamma}(\mathcal{X}|\mathcal{X}^{\text{old}}) \quad \text{s.t. } \mathcal{O}_{\Omega} = \mathcal{X}_{\Omega}. \quad (14)$$

Problem (14) is convex, thus we can use ADMM to solve it. Introducing auxiliary variable \mathcal{M} and let D be the feasible domain $\{\mathcal{X}|\mathcal{O}_{\Omega} = \mathcal{X}_{\Omega}\}$, then Equation (13) is equivalent to

$$\min_{\mathcal{M} \in D} Q_{\gamma}(\mathcal{M}|\mathcal{X}^{\text{old}}) \quad \text{s.t. } \mathcal{M} = \mathcal{X}. \quad (15)$$

Problem (15) is easy to solve by standard ADMM iterations, and the derivation of ADMM steps are omitted here for page limit and included in the supplemental material. Note that ADMM is the inner loop, after the ADMM converges we should update \mathcal{X}^{old} and repeat ADMM iterations again. Detailed algorithm is described in Algorithm 1. We have the following theoretical guarantee for this algorithm.

Theorem 5.1: The iteration sequence generated by $\mathcal{X}^{t+1} \in \arg \min_{\mathcal{O}_{\Omega} = \mathcal{X}_{\Omega}} Q_{\gamma}(\mathcal{X}|\mathcal{X}^t)$ is non-increasing, *i.e.*, $\|\mathcal{X}^{t+1}\|_{\gamma} \leq$

Algorithm 2 MM algorithm for tensor RPCA

Input: \mathcal{X} **Hyper parameters:** $\gamma_1, \gamma_2, \mu_0, \rho, \mu_{\max}$

- 1: Initialize $\mathcal{L}^0, \mathcal{E}^0$ by other tensor RPCA algorithm
 - 2: Initialize \mathcal{Y}^0 by random guess
 - 3: **while** not converged **do**
 - 4: Calculate t-SVD of $\mathcal{L}^{\text{old}} = \mathcal{U} * \mathcal{S}^{\text{old}} * \mathcal{V}^*$
 - 5: Set $\mathcal{Z}_{ijk}^t = \delta_i^j \varphi'_{1, \gamma_1}(\overline{\mathcal{S}}_{iik}^{\text{old}}) / \mu$
 - 6: Set $\mathcal{W}_{ijk}^t = \varphi'_{\lambda, \gamma_2}(\mathcal{E}_{ijk}^{\text{old}}) / \mu$
 - 7: **while** not converged **do**
 - 8: $\mathcal{L}_{l+1}^t = \mathcal{D}_{\mathcal{Z}^t}(\mathcal{X} - (\mathcal{E}_l^t + \frac{1}{\mu_l} \mathcal{Y}_l^t))$
 - 9: $\mathcal{E}_{l+1}^t = \mathcal{T}_{\mathcal{W}^t}(\mathcal{X} - (\mathcal{L}_{l+1}^t + \frac{1}{\mu_l} \mathcal{Y}_l^t))$
 - 10: $\mathcal{Y}_{l+1}^t = \mathcal{Y}_l^t + \mu_l \mathcal{L}_{l+1}^t + \mathcal{E}_{l+1}^t - \mathcal{X}$
 - 11: $\mu_{l+1} = \min(\rho \mu_l, \mu_{\max})$
 - 12: **end while**
 - 13: Update $\mathcal{L}^{t+1}, \mathcal{E}^{t+1}$ by the result of inner iteration
 - 14: Set $\mathcal{L}^{\text{old}} = \mathcal{L}^{t+1}, \mathcal{E}^{\text{old}} = \mathcal{E}^{t+1}$
 - 15: **end while**
-

$\|\mathcal{X}^t\|_\gamma$ and converges to some Q^* . Besides, there exists a subsequence $\{\mathcal{X}^{i_k}\}_{k=1}^\infty$ converges to a minimal point \mathcal{X}_* of $\|\mathcal{X}\|_\gamma$ on $\{\mathcal{X} | \mathcal{O}_\Omega = \mathcal{X}_\Omega\}$.

B. Non-convex Tensor Robust PCA

Given a tensor \mathcal{X} , the goal of robust PCA is to decompose \mathcal{X} into two parts: low-rank tensor \mathcal{L} and sparse tensor \mathcal{E} . This problem can be formulated as

$$\min_{\mathcal{L}, \mathcal{E}} \text{rank}(\mathcal{L}) + \|\mathcal{E}\|_0 \quad \text{s.t. } \mathcal{L} + \mathcal{E} = \mathcal{X}. \quad (16)$$

Apply the proposed novel sparsity measure and tensor γ -norm, we obtain

$$\min_{\mathcal{L}, \mathcal{E}} \|\mathcal{L}\|_{\gamma_1} + \Phi_{\lambda, \gamma_2}(\mathcal{E}) \quad \text{s.t. } \mathcal{L} + \mathcal{E} = \mathcal{X}. \quad (17)$$

We may set $\varphi_{\lambda, \gamma}$ to be SCAP or MCP in Equation (17), and refer them as TRPCA_{scad} and TRPCA_{mcp} respectively.

We apply majorization minimization algorithm to solve problem (17). Given $\mathcal{L}^{\text{old}}, \mathcal{E}^{\text{old}}$, we minimize the upper bound of $\|\mathcal{L}\|_{\gamma_1} + \Phi_{\lambda, \gamma_2}(\mathcal{E})$ given in Theorem 4.3,

$$\min_{\mathcal{L}, \mathcal{E}} Q_{\gamma_1}(\mathcal{L} | \mathcal{L}^{\text{old}}) + Q_{\lambda, \gamma_2}(\mathcal{E} | \mathcal{E}^{\text{old}}) \quad \text{s.t. } \mathcal{L} + \mathcal{E} = \mathcal{X}. \quad (18)$$

This problem is also easy to solve by ADMM. The sub-problem of updating \mathcal{L} and \mathcal{E} has closed-form solutions according to Theorem 4.4 and Theorem 4.5. We describe the detailed algorithm in Algorithm 2. We have the following theoretical result for this algorithm.

Theorem 5.2: The iteration sequence generated by

$$(\mathcal{L}^{t+1}, \mathcal{E}^{t+1}) \in \arg \min_{\mathcal{L} + \mathcal{E} = \mathcal{X}} Q_{\gamma_1}(\mathcal{L} | \mathcal{L}^t) + Q_{\lambda, \gamma_2}(\mathcal{E} | \mathcal{E}^t)$$

is non-increasing, *i.e.*, $\|\mathcal{L}^{t+1}\|_{\gamma_1} + \Phi_{\lambda, \gamma_2}(\mathcal{E}^{t+1}) \leq \|\mathcal{L}^t\|_{\gamma_1} + \Phi_{\lambda, \gamma_2}(\mathcal{E}^t)$ and converges to some Q^* . Besides, there exists a subsequence $\{(\mathcal{L}^{i_k}, \mathcal{E}^{i_k})\}_{k=1}^\infty$ converges to a minimal point $(\mathcal{L}_*, \mathcal{E}_*)$ of $\|\mathcal{L}\|_{\gamma_1} + \Phi_{\lambda, \gamma_2}(\mathcal{E})$ on $\{(\mathcal{L}, \mathcal{E}) | \mathcal{L} + \mathcal{E} = \mathcal{X}\}$.

VI. EXPERIMENTS

A. Datasets and Experimental Settings

We evaluate the effectiveness of the proposed non-convex tensor completion and tensor RPCA algorithms on Berkeley Segmentation 500 Dataset (BSD 500) [32] and Natural Scenes 2002 Dataset (NS 2002) [33]. Berkeley Segmentation 500 Dataset consists of 500 natural images, and Natural Scenes 2002 Dataset contains 8 hyperspectral images with 31 bands sampled from 410nm to 710nm at 10nm intervals. All the hyperspectral images are downsampled by factor 2. We employ Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Feature SIMilarity (FSIM) [34], Erreur Relative Globale Adimensionnelle de Synthèse (ERGAS)[35] and Spectral Angle Mapper (SAM) [36], [37] as performance evaluation indexes. Smaller MSE, ERGAS, SAM and larger PSNR, FSIM indicates the result is better.

There are some practical issues to clarify about Algorithm 1 and Algorithm 2. First, the hyper-parameters μ_0, ρ, μ_{\max} are introduced to accelerate the convergence speed. The inner ADMM iteration is always convergent regardless of the settings of these parameters, but the speed of convergence is different. In practice, we find setting $\mu_0 = 1, \rho = 1.1, \mu_{\max} = 1e10$ results in fast convergence. Second, the initialization of \mathcal{X}^0 and \mathcal{L}, \mathcal{E} is very important, since a good starting position usually leads to better final result in non-convex optimization problems. We suggest initializing \mathcal{X}^0 or $\mathcal{L}^0, \mathcal{E}^0$ by other tensor completion or tensor RPCA methods (such as TRPCA [23]). Last but not least, it usually takes a long time for the outer iteration to converge. In practice, it's not necessary to wait for convergence. Instead, we can iterate the outer loop for fixed times.

B. Tensor Completion Experiments

We conduct tensor completion experiments on BSD 500 and NS 2002 to test the performances of LRTC_{mcp} and LRTC_{scad}. For comparison, we also consider five competing tensor completion methods: Bayesian CP Factorization (FBCP) [3], Simple Low-Rank Tensor Completion (SiLRTC) [2], High Accuracy Low-Rank Tensor Completion (HaLRTC) [2], tensor-SVD based method (t-SVD) [21], twist Tensor Nuclear Norm based method (t-TNN) [11].

Natural image inpainting. We randomly select 200 images in BSD 500 for evaluation. For each image, pixels are randomly sampled with a sampling rate ranging from 20% to 80%. For our LRTC_{mcp} and LRTC_{scad} models, we set $\gamma = 25$, and use the result of t-TNN as initialization. To alleviate redundant computations, we apply the one-step LLA strategy [38], [24], *i.e.*, we run the outer loop only once instead of waiting for convergence. The average performances over selected 200 images under different sampling rates are summarized in Table I. From this table, we can see that our proposed LRTC_{mcp} and LRTC_{scad} outperform other competing methods in terms of all performance evaluation indices. As for efficiency, the proposed methods are significantly faster than FBCP, SiLRTC, HaLRTC, and t-SVD. Since the proposed methods are initialized by t-TNN, the running time is always slightly longer than t-TNN.



Fig. 3. Tensor completion performance comparison on example images.

TABLE I
TENSOR COMPLETION PERFORMANCES EVALUATION ON NATURAL IMAGES UNDER VARYING SAMPLING RATES.

Method	20%			40%			60%			80%			time (s)
	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM	
SiLRTC	23.59	0.798	0.822	27.987	0.899	0.915	32.24	0.951	0.964	37.47	0.977	0.988	19.95
HaLRTC	23.82	0.797	0.828	28.39	0.902	0.920	33.038	0.953	0.968	39.27	0.978	0.991	31.32
FBCP	24.08	0.668	0.794	26.40	0.753	0.837	27.35	0.799	0.857	27.71	0.82	0.865	103.68
t-SVD	24.13	0.764	0.835	29.703	0.893	0.931	36.03	0.950	0.977	45.04	0.969	0.992	33.47
t-TNN	25.30	0.841	0.864	30.50	0.923	0.943	36.27	0.952	0.978	44.14	0.967	0.991	3.03
LRTC _{mcp}	25.70	0.845	0.869	31.06	0.927	0.946	36.87	0.959	0.980	45.46	0.973	0.993	3.79
LRTC _{scad}	25.70	0.844	0.869	31.04	0.926	0.946	36.84	0.959	0.980	45.45	0.973	0.993	3.83

TABLE II
TENSOR COMPLETION PERFORMANCES EVALUATION ON HYPERSPECTRAL IMAGES UNDER VARYING SAMPLING RATES. THE UNIT IS 10^{-4} FOR MSE.

Method	20%			40%			60%			80%			time (s)
	PSNR	MSE	ERGAS	PSNR	MSE	ERGAS	PSNR	MSE	ERGAS	PSNR	MSE	ERGAS	
SiLRTC	41.71	4.70	30.912	45.46	1.95	21.524	49.14	0.827	14.412	52.86	0.354	10.341	42.21
HaLRTC	42.11	4.53	29.626	45.95	1.90	20.556	49.79	0.801	13.514	53.67	0.342	9.660	53.11
FBCP	37.09	14.47	52.931	43.25	3.85	29.318	46.00	2.225	22.344	46.67	2.011	20.688	210.18
t-SVD	41.64	5.10	31.835	45.52	2.12	22.142	49.42	0.886	14.685	53.49	0.365	10.157	224.21
t-TNN	42.46	3.81	28.702	46.07	1.60	20.135	49.82	0.667	13.272	53.61	0.290	9.515	47.29
LRTC _{mcp}	42.91	3.58	27.799	46.75	1.51	19.684	50.47	0.665	13.293	54.11	0.316	9.642	70.95
LRTC _{scad}	42.91	3.58	27.804	46.76	1.51	19.684	50.46	0.666	13.298	54.11	0.316	9.642	71.14

However, the performances are improved by only one MM iteration and the extra running times are marginal. Therefore, we claim that it is necessary to introduce non-convexity in the tensor completion task. These observations demonstrate LRTC_{mcp} and LRTC_{scad} are both effective and efficient. We also give visual comparisons in Figure 3.

Hyperspectral image inpainting. We use all 8 hyperspectral images in this experiment. For each hyperspectral image, we randomly sample its elements with a sampling rate ranging from 0.2 to 0.8. Since the sizes of hyperspectral images are relatively large, we run the outer loop in Algorithm 1 for 10 iterations based on t-TNN initialization. The performance comparison are shown in Table II. We have similar observations as in the natural image case: the results obtained by LRTC_{mcp} and LRTC_{scad} have lower MSE, ERGAS, and

higher PSNR, indicating that the proposed methods outperform competing methods. The fastest and the slowest algorithms are different in Table I and II mainly because $n_3 = 3$ for natural images while $n_3 = 31$ for hyperspectral images and the time complexities of these algorithms depends on n_3 differently. Besides, we find that the results of LRTC_{mcp} and LRTC_{scad} are nearly the same since MCP and SCAD shares very similar properties.

C. Tensor RPCA Experiments

We compare our proposed TRPCA_{mcp} and TRPCA_{scad} with matrix RPCA [9], [8] and TNN based TRPCA [10], [23] on both natural images and multispectral images. To apply matrix RPCA in tensor RPCA task, we simply apply matrix RPCA to each frontal slices of the corrupted tensor.

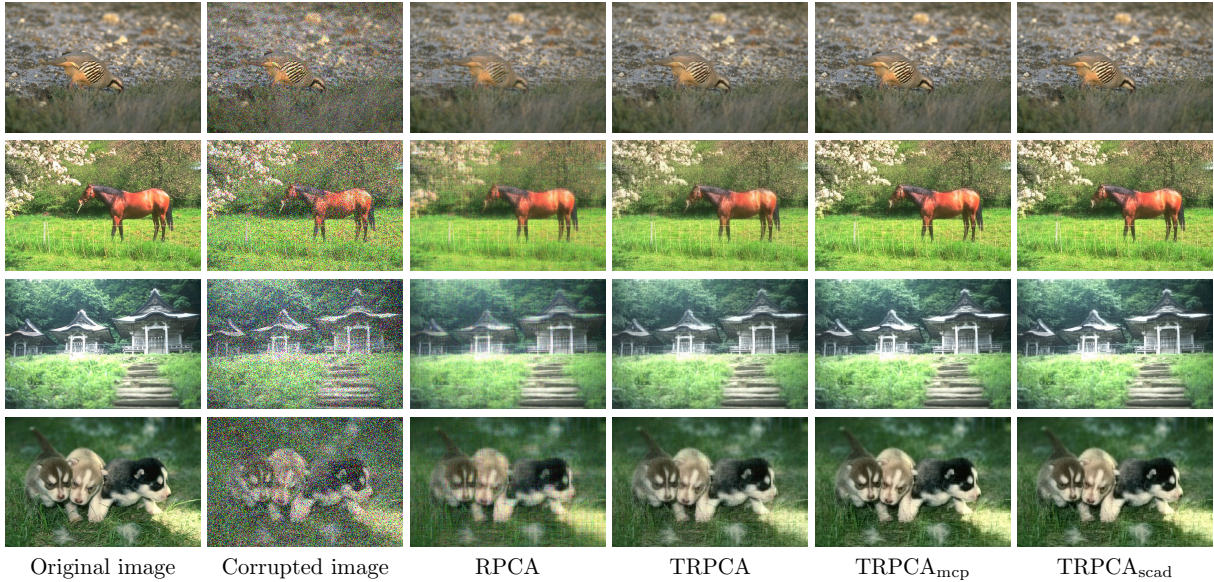


Fig. 4. Tensor RPCA performance comparison on example images. From top to bottom: $p_n = 0.1, 0.2, 0.3, 0.4$.

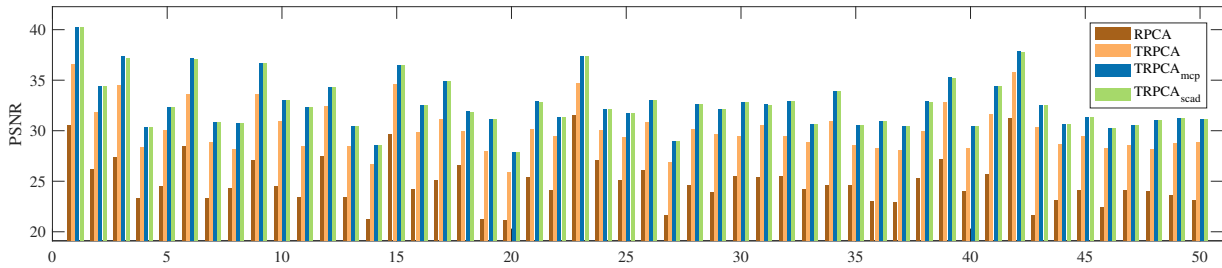


Fig. 5. Comparison of PSNR values obtained by RPCA, TRPCA, TRPCA_{mcp}, TRPCA_{scad} on randomly selected 50 images.

Natural image restoration. We first test TRPCA_{mcp} and TRPCA_{scad} on BSD 500. Each image is corrupted by salt-and-pepper noise with probability $p_n = 0.1$. We set $\gamma_1 = \gamma_2 = 20$ and run the outer loop of Algorithm 2 for 10 iterations. Performance comparison on randomly selected 50 images are shown in Figure 5. From this figure, we have following observations. First, the results of TRPCA, TRPCA_{mcp} and TRPCA_{scad} are significantly better than results of RPCA. This indicates that considering tensor structure helps to improve recovery quality compared to consider each channel individually. Second, TRPCA_{mcp} and TRPCA_{scad} obtained better performance than TRPCA, which means introducing concavity to Tensor RPCA tasks are necessary. Third, the PSNR values of TRPCA_{mcp} and TRPCA_{scad} are very similar, indicating the final result is not sensitive to the choice of non-convex penalty. We also give visual comparisons in Figure 4. Note that for the noise proportion ranging from 0.1 to 0.4 in Figure 4.

Hyperspectral image restoration. In this part, we test the proposed models on NS 2002. We add random noise to each hyperspectral image with probability p_n ranging from 0.1 to 0.4. Here, the noise is uniformly distributed in $[0, 0.1 * M)$ where M is the maximum absolute value of the original image. We set $\gamma_1 = \gamma_2 = 50$, and run the outer loop for

TABLE III
TENSOR ROBUST PRINCIPAL COMPONENT ANALYSIS PERFORMANCES
EVALUATION ON HYPER-SPECTRAL IMAGES.

p_n	Index	RPCA	TRPCA	TRPCA _{mcp}	TRPCA _{scad}
0.1	MSE	20.113	3.680	3.332	3.329
	PSNR	41.89	46.11	46.57	46.57
	ERGAS	23.224	13.203	12.898	12.893
	SAM	0.0887	0.0894	0.0880	0.0879
0.2	MSE	21.298	4.353	3.837	3.834
	PSNR	41.31	45.63	46.08	46.09
	ERGAS	24.374	14.473	13.703	13.706
	SAM	0.1191	0.1003	0.0961	0.0961
0.3	MSE	26.430	6.637	4.731	4.726
	PSNR	39.68	44.13	45.28	45.27
	ERGAS	28.956	18.668	15.177	15.205
	SAM	0.1710	0.1291	0.1076	0.1079
0.4	MSE	49.858	28.275	6.979	6.964
	PSNR	36.13	38.51	43.16	43.11
	ERGAS	49.877	50.354	20.126	20.298
	SAM	0.2602	0.2483	0.1428	0.1440

10 iterations. The results of TRPCA are used as initialization for the proposed methods. We employ MSE, PSNR, ERGAS, and SAM as quality indexes. The results are reported in

Table III. It's easy to see that $\text{TRPCA}_{\text{mcp}}$ and $\text{TRPCA}_{\text{scad}}$ outperform competing methods in terms of all quality indexes. Specifically, we note that when $p_n = 0.4$, *i.e.*, the noise proportion is rather large, the proposed methods improve the results of TRPCA significantly. In these circumstances, the sparse assumption on noise may not hold. Although RPCA and TRPCA have nice exact recovery property under certain conditions, these conditions are rather strict and sometimes not satisfied. However, the proposed methods still recover the images successfully.

VII. CONCLUSIONS

In this paper, we have presented a new non-convex tensor rank surrogate function and a new non-convex sparsity measure based on the MCP and SCAD penalty functions. Then, we have analyzed some theoretical properties of the proposed penalties. In particular, we applied the non-convex penalties in tensor completion and tensor robust principal component analysis tasks, and devised optimization algorithms based on majorization minimization. Experimental results on natural images and hyperspectral images substantiate the proposed methods outperform competing methods.

ACKNOWLEDGEMENT

This work was supported by the National Key Research and Development Program of China under grant 2018AAA0100205.

REFERENCES

- [1] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
- [2] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE TPAMI*, vol. 35, no. 1, pp. 208–220, 2013.
- [3] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian cp factorization of incomplete tensors with automatic rank determination," *IEEE TPAMI*, vol. 37, no. 9, pp. 1751–1763, 2015.
- [4] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.
- [5] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE TSP*, vol. 65, no. 13, 2017.
- [6] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, p. 717, 2009.
- [7] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.
- [8] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.
- [9] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *NeurIPS*, 2009.
- [10] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *IEEE CVPR*, 2016.
- [11] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang, "A new low-rank tensor model for video completion," *arXiv:1509.02027*, 2015.
- [12] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.
- [13] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *The Annals of statistics*, vol. 38, no. 2, 2010.
- [14] J. Shi, X. Ren, G. Dai, J. Wang, and Z. Zhang, "A non-convex relaxation approach to sparse dictionary learning," in *IEEE CVPR*, 2011.
- [15] Z. Zhang and B. Tu, "Nonconvex penalization using laplace exponents and concave conjugates," in *NeurIPS*, 2012.
- [16] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [17] D. R. Hunter and R. Li, "Variable selection using mm algorithms," *Annals of statistics*, vol. 33, no. 4, p. 1617, 2005.
- [18] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE TSP*, vol. 65, no. 3, pp. 794–816, 2017.
- [19] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, 2011.
- [20] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 148–172, 2013.
- [21] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-svd," in *IEEE CVPR*, 2014.
- [22] P. Zhou and J. Feng, "Outlier-robust tensor pca," in *IEEE CVPR*, 2017.
- [23] C. Lu, J. Feng, W. Liu, Z. Lin, S. Yan *et al.*, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE TPAMI*, 2019.
- [24] S. Wang, D. Liu, and Z. Zhang, "Nonconvex relaxation approaches to robust matrix recovery," in *IJCAI*, 2013.
- [25] W. Cao, Y. Wang, C. Yang, X. Chang, Z. Han, and Z. Xu, "Folded-concave penalization approaches to tensor completion," *Neurocomputing*, vol. 152, pp. 261–273, 2015.
- [26] T.-Y. Ji, T.-Z. Huang, X.-L. Zhao, T.-H. Ma, and L.-J. Deng, "A non-convex tensor rank approximation for tensor completion," *Applied Mathematical Modelling*, vol. 48, pp. 410–422, 2017.
- [27] Q. Zhao, D. Meng, X. Kong, Q. Xie, W. Cao, Y. Wang, and Z. Xu, "A novel sparsity measure for tensor recovery," in *IEEE ICCV*, 2015.
- [28] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, and L.-J. Deng, "A novel non-convex approach to recover the low-tubal-rank tensor data: when t-svd meets pssv," *arXiv:1712.05870*, 2017.
- [29] W.-H. Xu, X.-L. Zhao, T.-Y. Ji, J.-Q. Miao, T.-H. Ma, S. Wang, and T.-Z. Huang, "Laplace function based nonconvex surrogate for low-rank tensor completion," *Signal Processing: Image Communication*, 2018.
- [30] T. Yokota, B. Erem, S. K. Warfield, and H. Hontani, "Missing slice recovery for tensors using a low-rank model in embedded space," in *IEEE CVPR*, 2018.
- [31] Q. Yao, J. T.-Y. Kwok, and B. Han, "Efficient nonconvex regularized tensor completion with structure-aware proximal iterations," in *ICML*, 2019.
- [32] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE TPAMI*, vol. 33, no. 5, pp. 898–916, May 2011. [Online]. Available: <http://dx.doi.org/10.1109/TPAMI.2010.161>
- [33] S. M. Nascimento, F. P. Ferreira, and D. H. Foster, "Statistics of spatial cone-excitation ratios in natural scenes," *JOSA A*, vol. 19, no. 8, pp. 1484–1490, 2002.
- [34] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE TIP*, vol. 20, no. 8, pp. 2378–2386, 2011.
- [35] L. Wald, *Data fusion: definitions and architectures: fusion of images of different spatial resolutions*. Presses des MINES, 2002.
- [36] F. A. Kruse, A. Lefkoff, J. Boardman, K. Heidebrecht, A. Shapiro, P. Barloon, and A. Goetz, "The spectral image processing system (sips)—interactive visualization and analysis of imaging spectrometer data," *Remote sensing of environment*, vol. 44, no. 2-3, pp. 145–163, 1993.
- [37] R. H. Yuhua, A. F. Goetz, and J. W. Boardman, "Discrimination among semi-arid landscape endmembers using the spectral angle mapper (sam) algorithm," 1992.
- [38] H. Zou and R. Li, "One-step sparse estimates in nonconcave penalized likelihood models," *Annals of statistics*, vol. 36, no. 4, p. 1509, 2008.