Two New Low Rank Tensor Completion Methods Based on Sum Nuclear Norm

Hongbing Zhang*a, Xinyi Liu*a, Hongtao Fan*a,‡, Yajing Li*a, Yinlin Ye*a, Xinyun Zhu*b

*aCollege of Science, Northwest A&F University, Yangling, Shaanxi 712100, PR China
bDepartment of Mathematics, University of Texas of the Permian Basin, Odessa, TX 79762, USA

Abstract

The low rank tensor completion (LRTC) problem has attracted great attention in computer vision and signal processing. How to acquire high quality image recovery effect is still an urgent task to be solved at present. This paper proposes a new tensor $L_{2,1}$ norm minimization model (TLNM) that integrates sum nuclear norm (SNN) method, differing from the classical tensor nuclear norm (TNN)-based tensor completion method, with $L_{2,1}$ norm and Qatar Riyal decomposition for solving the LRTC problem. To improve the utilization rate of the local prior information of the image, a total variation (TV) regularization term is introduced, resulting in a new class of tensor $L_{2,1}$ norm minimization with total variation model (TLNMTV). Both proposed models are convex and therefore have global optimal solutions. Moreover, we adopt the Alternating Direction Multiplier Method (ADMM) to obtain the closed-form solution of each variable, thus ensuring the feasibility of the algorithm. Numerical experiments show that the two proposed algorithms are convergent and outperform compared methods. In particular, our method significantly outperforms the contrastive methods when the sampling rate of hyperspectral images is 2.5%.

Keywords: Low rank tensor completion, sum nuclear norm (SNN) method, Qatar Riyal decomposition, $L_{2,1}$ norm, total variation, alternating direction multiplier method.

1. Introduction

The low rank tensor completion (LRTC) problem, especially when dealing with extremely high-dimensional data, such as appearing in color image and video processing [1], [2], magnetic resonance image [3], [4], hyperspectral image [5], [6], [7], pattern recognition [8], [9] face modeling and analysis [10] and other fields, has received extensive attention. The fact that the imaging of visual data is based on poor acquisition conditions or severe data corruption during transmission, results in incomplete or severely corrupted data being acquired. This will facilitate LRTC [11] to become crucially important...
for collecting data, particularly how to utilize the internal structural information between collected observations and missing data to achieve LRTC problem.

Mathematically, the above LRTC problems can be can be characterized as follows:

$$\min_{\mathcal{X}} \text{rank}(\mathcal{X}), \quad \text{s.t.} \quad \mathcal{X}_\Omega = \mathcal{T}_\Omega$$  \hspace{1cm} (1)$$

where $T$ and $\mathcal{X}$ represent the incomplete and complete tensors, respectively, $\Omega$ denotes an index set of observed data of $T$, $\text{rank}(\mathcal{X})$ implies tensor rank of any tensor $\mathcal{X}$, conditions to be satisfied $\mathcal{X}_\Omega = \mathcal{T}_\Omega$ mean that the elements at the same positions in two sets $\mathcal{X}$ and $\mathcal{T}$ remain equal according to index set $\Omega$.

It is well-known, on the one hand, that the definition of tensor rank is not unique, and its various definitions of tensor rank have been proposed according to different ways of tensor decomposition, such as CANDECOMP/PARAFAC(CP) rank [12], [13] generated by CP decomposition [14], Tucker-rank established by Tucker decomposition [15], [16], and tube rank and multi-rank based on tensor products [17], [18], etc. However, specific features, such as the calculation of the rank of a tensor CP is NP-hard [19], unfolding and folding of mode-$n$ on a tensor in Tucker rank [20], [21] lead to dimensional disaster, and the limitation of the tensor product definition results in the multi-rank and tube rank [22], [23] based on t-SVD decomposition are only applicable to third-order tensors, indicate that the definitions of different tensor rank show their respective inadequacies. The other half is that tensor rank is a non-convex function.

In this regard, Liu et al. [24] turned to the convex envelope nuclear norm of the rank function to approximate it, and proposed a tensor nuclear norm minimization method to solve such problems, specifically, for image recovery problems

$$\min_{\mathcal{X}} \|\mathcal{X}\|_*, \quad \text{s.t.} \quad \mathcal{X}_\Omega = \mathcal{T}_\Omega$$  \hspace{1cm} (2)$$

where $\|\mathcal{X}\|_* = \sum_{n=1}^{N} \alpha_n \|\mathcal{X}(n)\|_*$ with $\alpha_n \geq 0$ and $\sum_{n=1}^{N} \alpha_n = 1$.

To further enhance, despite the model (2) has ideal effect in global information recovery, the performance of the model for local information extraction, especially in processing high-dimensional image recovery, by incorporating local piece-wise smoothness prior, and proposed a large class of Total Variation (TV) minimization methods [25] and applied to image processing and pattern recognition [16], [26], [27], [28]. In fact, for LRTC problems, TV terms are usually included in a low-rank (LR) framework to jointly represent local piecewise continuity and global LR structure along different dimensions [29], [30], [31], [32], which can greatly improve the local information effect of the model. Its specific modeling is as follows:
\[
\min_{\mathcal{X}} \|\mathcal{X}\|_* + \lambda \|\mathcal{X}\|_{TV}, \quad s.t. \quad \mathcal{X}_{\Omega} = T_{\Omega}
\]

where \(\|\mathcal{X}\|_{TV}\) is TV regularization term, and \(\lambda > 0\) is a parameter used to balance the relationship between the nuclear norm and the regularization term.

Recently, Liu et al. [33] proposed an approximate SVD computation technique based on QR decomposition (CSVD-QR), and integrated L2,1norm to construct a fast and accurate matrix completion method. Inspired by the high performance of this method, Zheng et al. [34] designed a numerical method that approximates t-SVD combined with Qatari Riyal Decomposition (CTSVD-QR), which further generalizes the matrix-form QR method to the tensor case. Through in-depth comparative analysis and thorough experimental exploration, it is found that although the speed of the algorithm has been significantly improved, compared with the classical tensor nuclear norm (TNN) method, the performance of the algorithm in terms of image restoration effect or quality is still less ideal. In view of this, to improve and develop this method, we consider integrating tensor SNN, not TNN, with QR decomposition and \(L^2_{2,1}\) norm to design and build a more efficient and feasible method, namely, tensor \(L^2_{2,1}\) norm minimization model (TLNM). In order to further improve the effect of image restoration, that is, to better protect the local detail information while better representing the global information of the image, a TV regularization term is introduced and then a class of tensor \(L^2_{2,1}\) norm minimizes the total variation model is proposed. To sum up, our main contributions are as follows:

Firstly, we adopt a new method different from TNN, that is, the integration of SNN and CSVD-QR method, to improve and enhance the restoration effect of the combination of LRTC and CSVD-QR method. For the sake of fairness, we still choose the \(L^2_{2,1}\) norm instead of the classical nuclear norm in the objective function as in the recently published papers [33] and [34]. In the meantime, by introducing the TV regularization term into the model constructed above, a new method named tensor \(L^2_{2,1}\) norm minimization with TV (TLNMTV) is established to better preserve some details of the image, thereby improving the utilization of the local prior information of the image.

Secondly, for the above two established LRTC models, namely TLNM and TLMNTV, two efficient alternating direction multiplier (ADMM) algorithms [35], [36] are designed to solve them respectively. The closed-form solution of each variable can be obtained by the ADMM algorithm, so that the algorithm can be effectively implemented, thus maintaining the accuracy. Furthermore, the experimental results show that the relative error (RE) of these two algorithms always decreases, hence ensuring that they are indeed convergent.

Thirdly, extensive real-data experiments show that the TLNM method obviously outperforms the classical TNN method, which indicates that our proposed integrating QR decomposition and SNN method outperforms the TNN-based TLNM-TQR [34] method. More importantly, the TLMNTV
method further enhances the local sparsity, with higher quantitative numerical result and better visual restoration effect than other comparison methods in this paper. It is also particularly worth mentioning that when the sampling rate of hyperspectral images is 2.5%, our methods can perform image restoration clearly and very well and are shown to be the best.

2. PRELIMINARIES

We are now in a position to give the basic notation of the tensors involved and lay out the basic definitions and theorems for constructing the two proposed new methods.

2.1. Tensor Notations and their Definitions

Generally, the uppercase and lowercase of any letter represent respectively vectors and matrices, such as \( x \) represents a vector and \( X \) represents a matrix. We use a calligraphic upper case letter \( \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \) to represent an Nth-order tensor, and \( x_{i_1, i_2, \ldots, i_N} \) to represent its \((i_1, i_2, \ldots, i_N)\)-th element. Unless otherwise stated, we still use \( \|X\|_F = \left( \sum_{i_1, i_2, \ldots, i_N} x_{i_1, i_2, \ldots, i_N}^2 \right)^{1/2} \) as the definition of the Frobenius norm and define the inner product of two Nth-order tensors \( Y \) and \( Z \) as \( \langle Y, Z \rangle = \sum_{i_1, i_2, \ldots, i_N} y_{i_1, i_2, \ldots, i_N} z_{i_1, i_2, \ldots, i_N} \), where \( y_{i_1, i_2, \ldots, i_N} \) and \( z_{i_1, i_2, \ldots, i_N} \) stand for the \((i_1, i_2, \ldots, i_N)\)-th element of \( Y \) and \( Z \), respectively.

Definition 1 ([37]). The mode-\( n \) unfolding of a tensor \( \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \) is denoted as a matrix \( \mathbf{X}(n) \in \mathbb{R}^{I_n \times I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N} \). Tensor element \((i_1, i_2, \ldots, i_N)\) maps to matrix element \((i_n, j)\), where

\[
j = 1 + \sum_{k=1, k \neq n}^{N} (i_k - 1)J_k \quad \text{with} \quad J_k = \prod_{m=1, m \neq n}^{k-1} I_m. \quad (4)
\]

The mode-\( n \) unfolding operator and its corresponding inverse operator are abbreviated as \( \text{unfold}_n \) and \( \text{fold}_n \), and satisfy transformation relation \( \mathcal{X} = \text{fold}_n(\mathbf{X}(n)) = \text{fold}_n(\text{unfold}_n(\mathcal{X})) \).

Definition 2 ([37]). The mode-\( n \) product operation between tensor \( \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \) and matrix \( U \in \mathbb{R}^{I_n \times I_{n+1}} \) is expressed as \( \mathcal{Y} = \mathcal{X} \times_n U \), where \( \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N} \). Elementwisely, we have

\[
\mathcal{Y} = \mathcal{X} \times_n U \iff \mathbf{Y}(n) = U \cdot \text{unfold}_n(\mathbf{X}(n)). \quad (5)
\]

Definition 3 ([38]). The \( L_{2,1} \) norm of a matrix \( M \in \mathbb{R}^{I \times J} \) can be defined as

\[
\|M\|_{2,1} = \sum_{j=1}^{J} \sqrt{\sum_{i=1}^{I} m_{i,j}^2}, \quad (6)
\]

where \( m_{i,j} \) represents the element at row \( i \)-th and column \( j \)-th of the matrix \( M \).
Algorithm 1 CSVD-QR\textsuperscript{33}

**Input:** $X$, a real matrix $C \in \mathbb{R}^{m \times n}$; $\text{It}_{\text{max}} > 0$

**Initialization:** $r > 0$, $q > 0$, $k = 1$; $\varepsilon$ is a positive tolerance, $L_0 = \text{eye}(m,r)$, $D_0 = \text{eye}(r,r)$, $R_0 = \text{eye}(r,n)$.

while $\|L_k D_k R_k - X\| \geq \varepsilon$ or $k < \text{It}_{\text{max}}$ do

$[Q, T] = \text{qr}(XR_k^T)$;

$L_{k+1} = Q(:, 1 : r)$.

$[Q, T] = \text{qr}(X^T L_{j+1})$;

$R_{k+1} = Q(:, 1 : r)^T$.

$D_{k+1} = T(1 : r, 1 : r)^T$.

$k = k + 1$.

end while

**return** $L = L_k$, $D = D_k$, $R = R_k$

**Output:** $L$, $D$, $R$ ($X = LDR$)

The $L_{2,1}$ norm has been successfully applied to low-rank representation \textsuperscript{35} to optimize the noise data matrix $K \in \mathbb{R}^{m \times n}$, which can be updated by solving the minimization problem as follows:

$$\min_K \tau \|K\|_{2,1} + \frac{1}{2} \|K - C\|_F^2,$$

where $C \in \mathbb{R}^{m \times n}$ is a pre-determined real matrix and $\tau > 0$.

Here we will describe an approximate SVD (CSVD-QR) method based on QR decomposition mainly used in this paper, see \textsuperscript{33}, and the corresponding CSVD-QR algorithm can be specifically shown in Algorithm 1 below.

**Theorem 1 ($L_{2,1}$ norm minimization solver (LNMS) \textsuperscript{33}).** The optimal $K(:,j)$, i.e., the $j$th column of $K$, of the problem in (7) follows

$$K(:,j) = \max\{\|C(:,j)\|_2 - \tau, 0\} \frac{C(:,j)}{\|C(:,j)\|_2},$$

where $\|C(:,j)\|_2 = \sqrt{\sum_{i=1}^{m} C_{ij}^2}$.

3. Models and Algorithms

3.1. Tensor $L_{2,1}$ Norm Minimization Model With Its Algorithm

For LRTC problem, to further improve the quality and visual effect of image restoration, we integrate the $L_{2,1}$ norm and CSVD-QR method into the tensor SNN minimization problem instead of the classical tensor TNN method, resulting in a tensor $L_{2,1}$ norm minimization model as below:
\[
\begin{align*}
\min_{\mathcal{X}} & \sum_{n=1}^{N} \alpha_n \|D_n\|_{2,1} \\
\text{s.t.} & \{M_n = \mathcal{X}\}_{n=1}^{N}, \quad \{M_{n(n)} = L_n D_n R_n\}_{n=1}^{N}, \\
\mathcal{X}_1 &= T_1, \quad L_n^T L_n = I_n, \quad R_n R_n^T = I_n,
\end{align*}
\]

(9)

where \(M_n, n = 1, 2, \ldots, N\) indicate auxiliary parameters, \(\sum_{n=1}^{N} \alpha_n = 1, L_n, D_n, R_n\) are induced by \(M_{n(n)}\) when completing CSVD-QR decomposition, \(\Omega\) represents the exact locations of known observations. It is worth mentioning here that we still adopt the \(L_2,1\) norm accommodating with CSVD-QR method and the tensor SNN minimization method to replace the nuclear norm in the objective function, which exactly shows the novelty and superiority of proposed method compared with the existing state-of-the-art methods and ensures the fairness in performing trial comparisons between two recently published methods [33] and [34].

3.2. Optimization TLNM Algorithm Based on ADMM

With the augmented Lagrangian formula, the above model (9) can be transformed into the following optimization form:

\[
\begin{align*}
\text{Lag}(\mathcal{X}, \{M_n, Q_n, L_n, R_n, D_n, \Phi_n\}_{n=1}^{N})
= & \sum_{n=1}^{N} \alpha_n \|D_n\|_{2,1} + \frac{\mu}{2} \|M_{n(n)} - L_n D_n R_n + \Phi_n\|_F^2 \\
+ & \frac{\mu}{2} \|M_n - \mathcal{X} + \frac{Q_n}{\mu}\|_F^2,
\end{align*}
\]

(10)

where matrices \(Q_n\) and \(\Phi_n\) denote the Lagrange multipliers and \(\mu > 0\). Now, we start to optimize the problem (10) under the ADMM framework. Here we use these matrix symbols \(L_1^+, R_1^+, D_1^+, M_1^+, \mathcal{X}^+\) to represent the final results of iterative updates in ADMM respectively.

3.2.1. Optimize \(\{L_1, \ldots, L_N\}\)

Let the initial values of all parameter matrices \(L_n, D_n, R_n\) in the proposed algorithm be \(L_n^0 = \text{eye}(I_n, r_n), D_n^0 = \text{eye}(r_n, r_n), R_n^0 = \text{eye}(r, t_n)\) and \(t_n = I_1 \cdots I_{n-1} I_{n+1} \cdots I_N\), respectively, where \text{eye} is the built-in command for the identity matrix in MATLAB. Keep other variables unchanged in (10), the optimization objective function with respect to \(L_n\) becomes

\[
\min_{L_n, R_n} \|M_{n(n)} - L_n D_n R_n + \frac{\Phi_n}{\mu}\|_F^2.
\]

(11)

Here, we regard the solution of problem (11) as the iterative solution process of the CSVD-QR method based on matrix \(M_{n(n)} + \frac{\Phi_n}{\mu}\), which can be accurately expressed as follows:

\[
M_{n(n)} + \frac{\Phi_n}{\mu} = L_n D_{Tn} R_n,
\]

(12)
where \( D_{Tn} \in \mathbb{R}^{r_n \times r_n} \). If we let \( M_{n(n)}^k + \frac{\Phi_n}{\mu} = G_n \), then \( L_{n}^{k+1} \) can be obtained recursively as follows:

\[
\begin{align*}
[Q, T] &= qr(G_n R_T^n) \\
L_n^+ &= Q(q_1, ..., q_{r_n})
\end{align*}
\]

(13)

(14)

where matrices \( Q \in \mathbb{R}^{I_n \times I_n} \) and \( T \in \mathbb{R}^{I_n \times r_n} \) are treated as intermediate variables.

3.2.2. Optimize \( \{ R_1, ..., R_N \} \)

In a similar manner, \( R_{n}^{k+1} \) can be derived iteratively as follows:

\[
\begin{align*}
[Q, T] &= qr(G_T^n L_n) \\
R_n^+ &= Q(q_1, ..., q_{r_n})^T
\end{align*}
\]

(15)

(16)

where \( Q \in \mathbb{R}^{I_n \times I_n} \) and \( T \in \mathbb{R}^{I_n \times r_n} \) are considered as intermediate variables.

3.2.3. Optimize \( \{ D_1, ..., D_N \} \)

With \( M_{n(n)}, \Phi_n, L_n, \) and \( R_n \) held fixed, the variable \( D_n \) can be optimally solved in the following way:

\[
D_n = \arg \min_{D_n} \frac{\alpha_n}{\mu} \| D_n \|_{2,1} + \frac{1}{2} \| D_n - L_n^T(M_{n(n)} + \frac{\Phi_n}{\mu})R_n^T \|^2_F.
\]

(17)

Combine (12) with (17), we get:

\[
D_{Rn} = L_n^T(M_{n(n)} + \frac{\Phi_n}{\mu})R_n^T
\]

(18)

As a result, (17) can be equivalently expressed as:

\[
D_n = \arg \min_{D_n} \frac{\alpha_n}{\mu} \| D_n \|_{2,1} + \frac{1}{2} \| D_n - D_{Rn} \|^2_F.
\]

(19)

With the result of Theorem [1] we can approximate and update \( D_n \) through the following formula:

\[
D_n^+ = D_{Rn} G_n,
\]

(20)

where

\[
G_n = diag(g_{n1}, ..., g_{nr_n}),
\]

(21)

and the jth entry \( g_{nj} \) can be calculated by using the following formula:

\[
g_{nj} = \frac{\max\{\| D_{Rn}(::j) \|_F - \frac{\alpha_n}{\mu}, 0\}}{\| D_{Rn}(::j) \|_F}.
\]

(22)
Algorithm 2 TLNM

Input: any incomplete tensor $\mathcal{T}$, the index set representing the position of known tensor elements $\Omega$, convergence criteria $\epsilon$, prescribed iteration number $K$.

Initialization: $\mathcal{X}^0 = \mathcal{T}_\Omega$, $\{\mathcal{M}^0_n = \mathcal{X}^0\}^N_{n=1}$, $\mu^0 > 0$, $\rho > 1$, $\{\alpha_n, r_n, I^0_n, D^0_n, R^0_n\}^N_{n=1}$.

1: while not converged and $k < K$ do
2: for $n=1:N$ do
3: Compute $L^k_n$ via (13-14); 
4: Compute $R^k_n$ via (15-16); 
5: Compute $D^k_n$ via (20); 
6: Compute $\mathcal{M}^k_k$ via (25); 
7: end for 
8: Compute $\mathcal{X}^k$ via (26); 
9: Compute the multipliers $Q^k_n$ and $\Phi^k_n$ via (27); 
10: $\mu^k = \rho\mu^{k-1}$, $k = k + 1$; 
11: Check whether the convergence condition $\|\mathcal{X}^{k+1} - \mathcal{X}^k\|_\infty \leq \epsilon$ is satisfied 
12: end while
13: return $\mathcal{X}^{k+1}$

Output: Completed tensor $\mathcal{X} = \mathcal{X}^{k+1}$

3.2.4. Optimize $\{\mathcal{M}_1, ..., \mathcal{M}_N\}$

Under the assumption that other variables remain unchanged, the optimization objective function with respect to $\mathcal{M}_n$ can degenerate into

$$\mathcal{M}_n = \min_{\mathcal{M}_n} \|\mathcal{M}_n(n) - L_n D_n R_n + \frac{\Phi_n}{\mu} \|_F^2 \right.$$  
$$+ \|\mathcal{M}_n(n) - \mathcal{X} + \frac{Q_n(n)}{\mu} \|_F^2. \tag{23}$$

Using the definitions and properties of tensor (matrix) Frobenius norm, $\mathcal{M}_n$ of (23) can be equivalently reformulated as

$$\mathcal{M}_n(n) = \min_{\mathcal{M}_n} \|\mathcal{M}_n(n) - L_n D_n R_n + \frac{\Phi_n}{\mu} \|_F^2 \right.$$  
$$+ \|\mathcal{M}_n(n) - \mathcal{X} + \frac{Q_n(n)}{\mu} \|_F^2. \tag{24}$$

Accordingly, $\mathcal{M}_n^+$ is deduced as follows:

$$\mathcal{M}_n^+ = fold_n\left(\frac{1}{2}(\mathcal{X} + \frac{Q_n(n)}{\mu} + L_n D_n R_n - \frac{\Phi_n}{\mu})\right). \tag{25}$$
3.2.5. **Optimize** \(X\)

The optimal solution for the variable \(X\) can be obtained by optimizing the minimization function of the following problem:

\[
X^+ = \min_X \sum_{n=1}^{N} \frac{\mu}{2} \|M_n - X + \frac{Q_n}{\mu}\|_F^2 \quad \text{s.t.} \quad X_{\Omega} = T_{\Omega}.
\]

its specific explicit closed-form solution can be written as

\[
\begin{cases}
X^+_{\Omega} = T_{\Omega}, \\
X^+_{\Omega^c} = \frac{1}{N\mu}(\sum_{n=1}^{N} \mu M_n + Q_n)_{\Omega^c},
\end{cases}
\]

where the vertical symbol \(\perp\) in the upper right corner of \(\Omega\) indicates the complement set \(\Omega^\perp\) of \(\Omega\).

3.2.6. **Update the multipliers** \(Q_n, \Phi_n\)

\[
\begin{cases}
Q^+_n = Q_n + \mu(M_n - X), \\
\Phi^+_n = \Phi_n + \mu(M_n(n) - L_D L_R), \\
\mu^+ = \rho \mu,
\end{cases}
\]

where \(\rho \geq 1\).

So far, all the above variables have been updated and solved, and the pseudocode corresponding to all algorithms for TLNM has been reported in detail in Algorithm 2.

3.3. **Tensor L\(_2,1\)-Norm Minimization with Total variation (TLNMTV)**

Total variation method is inherently remarkable in the protection of image details so as to improve the utilization rate of the local prior information of the image. In this regard, we investigate and propose the tensor \(L_{2,1}\) norm minimization with TV (TLNMTV) method to obtain the following model by adding the regularization term of the form in [31].

\[
\min_X \sum_{n=1}^{N} \alpha_n \|X_{(n)}\|_* + \lambda \sum_{n=1}^{N} \beta_n |C_n X_{(n)}| \quad \text{s.t.} \quad X_{\Omega} = T_{\Omega}.
\]

Here, the latter term is the TV regularization term, where the dimension of \(C_n\) is \((I_n - 1, I_n)\) with \((C_n)_{i,i} = 1\) and \((C_n)_{i,i+1} = -1\). The operator \(|\cdot|\) of any matrix \(X\) is defined as \(|X| = \sum_{i=1}^{I_n} \sum_{j=1}^{I_n} |X_{i,j}|\). The parameter \(\lambda\) is represented as a tunable hyperparameter. These parameters \(\beta_1, \ldots, \beta_N\) take either the value 0 or 1, which means whether the smoothing and piecewise prior information on the \(n\)-th mode of the completed tensor is known or not. However, as we know, different dimensions of different high-dimensional data represent different data information, and some information is not smooth. For this, it is not necessary to compute smoothing and piecewise priors in this dimension. It should be pointed
out that, quite different from the traditional regular meaning, the TV term used here is only able to
determine whether smoothing and piecewise priors need to be computed for different modes. Therefore,
it is more reasonable to use the TV regularization term in combination with the actual situation. In
particular, when the TV regularization term is removed, the problem degenerates itself into a low-rank
tensor problem \[ (2). \]

In this connection, by introducing auxiliary matrix and tensor variable sets \( \{ Q_n \}_{n=1}^N, \{ A_n \}_{n=1}^N \)
\( \{ Z_n \}_{n=1}^N \) for these variables \( X_{(n)}, C_nA_n \) and \( \chi \), respectively, i.e., \( A_n = X_{(n)}, \)
\( Q_n = C_nA_n \) and \( Z_n = \chi \), the proposed tensor \( L_{2,1}\)-norm minimization with TV model (TLNMTV) can be induced as
follows:

\[
\begin{align*}
\min_{\chi} & \sum_{n=1}^N \alpha_n \| D_n \|_{2,1} + \lambda \sum_{n=1}^N \beta_n |Q_n| \\
\text{s.t.} & \quad \chi = \mathcal{T}_\Omega, \{ Q_n = C_nA_n, Z_n = \chi, A_n = X_{(n)} \}_{n=1}^N, \\
& \quad \{ Z_{n(n)} = L_nD_nR_n \}_{n=1}^N, L_n^TL_n = I_n, R_n^TR_n = I_n. \tag{29}
\end{align*}
\]

Due to the fact that this model (29) is convex we can solve it using the ADMM framework.

### 3.4. Optimization TLNMTV Algorithm Based on ADMM

With the augmented Lagrange thinking, the optimization problem (29) can be transformed into:

\[
\begin{align*}
\text{Lag}(\chi, \{ Z_n \}, \{ Q_n \}, A_n, \{ A_n \}, \{ C_n \}, \{ L_n \}, \{ R_n \}, \{ D_n \}, \{ \Phi_n \}_{n=1}^N) &= \sum_{n=1}^N (\alpha_n \| D_n \|_{2,1} + \frac{\mu}{2} \| Z_n - \chi \|_F^2) + \frac{\mu}{2} \| Q_n - C_nA_n + \frac{\Lambda_n}{\mu} \|_F^2) \\
& \quad + \sum_{n=1}^N \beta_n (\lambda |Q_n| + \frac{\Lambda_n}{\mu} \| G_n \|_F^2) \quad \text{s.t.} \\
& \quad \frac{\mu}{2} \| A_n - X_{(n)} \|_F^2 + \frac{\mu}{2} \| F_n \|_F^2) \quad \\
& \quad + \sum_{n=1}^N \frac{\mu}{2} \| Z_{n(n)} - L_nD_nR_n + \frac{\Phi_n}{\mu} \|_F^2), \tag{30}
\end{align*}
\]

where matrices \( \{ A_n \}_{n=1}^N, \{ C_n \}_{n=1}^N, \{ \Phi_n \}_{n=1}^N \) and tensors \( \{ G_n \}_{n=1}^N \) represent, respectively, Lagrange multipliers, \( \mu > 0 \). Next, we explicitly give their respective formulas for these optimization variables
\( \{ Q_n \}_{n=1}^N, \{ L_n \}_{n=1}^N, \{ R_n \}_{n=1}^N, \{ D_n \}_{n=1}^N, \{ Z_n \}_{n=1}^N, \{ A_n \}_{n=1}^N \) and \( \chi \).

#### 3.4.1. Optimize \( \{ Q_1, \cdots, Q_N \} \)

Under the premise of ensuring that other variables do not change, we can transform the optimization
problem of \( \{ Q_1, \cdots, Q_N \} \) into a form of:

\[
\min_{\{ Q_1, \cdots, Q_N \}} \sum_{n=1}^N \beta_n (\lambda |Q_n| + \frac{\mu}{2} \| Q_n - C_nA_n + \frac{\Lambda_n}{\mu} \|_F^2). \tag{31}
\]
Based on the fact that the matrix groups \( \{Q_1, \cdots, Q_N\} \) in the optimization problem are independent of each other, the solution formula for each of them can be easily obtained:

\[
Q_n = \beta_n \cdot \text{shrinkage}_{\mu}(C_n A_n - \frac{1}{\mu} \Lambda_n),
\]

(32)

where the operator symbol \( \text{shrinkage}_{\alpha}(\cdot) \) represents the elementwise shrinkage thresholding operator of a matrix, i.e.,

\[
\text{shrinkage}_{\alpha}(X)_{i,j} = (X)_{i,j} - \min(\alpha, |(X)_{i,j}|) \cdot \frac{|(X)_{i,j}|}{|(X)_{i,j}|},
\]

and \( \frac{(X)_{i,j}}{|(X)_{i,j}|} \) is equal to zero when \( (X)_{i,j} = 0 \).

3.4.2. Optimize \( \{L_1, \ldots, L_N\}, \{R_1, \ldots, R_N\}, \{D_1, \ldots, D_N\} \)

Since the update of these variables \( L_1, \ldots, L_N, R_1, \ldots, R_N \) and \( D_1, \ldots, D_N \) is consistent with the TLNM algorithm, it is omitted here and will not be repeated.

3.4.3. Optimize \( \{Z_1, \ldots, Z_N\} \)

By keeping the rest of the variables unchanged, the optimization problem with respect to \( Z_n \) can be precisely described as below

\[
Z_n = \min_{Z_n} \|Z_n - L_n D_n R_n + \frac{\Phi_n}{\mu}\|_F^2 + \|Z_n - X + \frac{Q_n}{\mu}\|_F^2.
\]

(33)

With the help of the definition of tensor Frobenius norm and matrix Frobenius norm, the above problem (33) can be equivalently transformed into

\[
Z_n(n) = \min_{Z_n} \|2Z_n - L_n D_n R_n + \frac{\Phi_n}{\mu} - X(n) + \frac{Q_n(n)}{\mu}\|_F^2.
\]

(34)

Therefore \( Z_n^+ \) is obtained as follows:

\[
Z_n^+ = \text{fold}_n\left(\frac{1}{2\mu}(\mu X(n) - Q_n(n) + \mu L_n D_n R_n - \Phi_n)\right).
\]

(35)

3.4.4. Optimize \( \{A_1, \cdots, A_N\} \)

Make sure that the rest of the above variables do not change, we can reformulate the above optimization problem of \( \{A_1, \cdots, A_N\} \) into the following form:

\[
\min \sum_{n=1}^N \|Q_n - C_n A_n + \frac{A_n}{\mu}\|_F^2 + \|A_n - X(n) + \frac{\Gamma_n}{\mu}\|_F^2.
\]

(36)

This type of problem (36) can be solved by dividing it into \( N \) sub-problems, each of which can be expressed in this form:

\[
A_n = \arg \min_{A_n} \|Q_n - C_n A_n + \frac{A_n}{\mu}\|_F^2 + \|A_n - X(n) + \frac{\Gamma_n}{\mu}\|_F^2.
\]

(37)
In this regard, by solving the above minimization problem, the following update formula is derived:

\[
A_n^+ = (\mu C_n^T C_n + \mu I)^{-1}(C_n^T \Lambda_n + \mu C_n^T Q_n + \mu X_{(n)} - \Gamma_n),
\]

(38)

where \(I\) is the identity matrix of the appropriate order.

### 3.4.5. Optimize \(X\)

We obtain the variable \(X\) by solving the following constraint minimization function:

\[
X = \min \sum_{n=1}^{N} \left(\|Z_n - X + \frac{G_n}{\mu}\|_F^2 + \beta_n \|A_n - X_{(n)} + \frac{\Gamma_n}{\mu}\|_F^2\right)
\]

\[
s.t. \ X_{\Omega} = T_{\Omega},
\]

(39)

whose closed-form solution can be expressed as

\[
\begin{cases}
X_O^+ = T_{\Omega}, \\
X_{\Omega^+} = \frac{\left[\sum_{n=1}^{N} \mu Z_n + G_n + fold_n(\mu A_n + \Gamma_n)\right]_{\Omega^+}}{(N + \sum_{n=1}^{N} \beta_n) \mu}.
\end{cases}
\]

(40)

### 3.4.6. Update the multipliers \(G_n, \Lambda_n, \Phi_n\) and \(\Gamma_n\)

\[
\begin{cases}
G_n^+ = G_n + \mu(Z_n - X), \\
\Lambda_n^+ = \Lambda_n + \mu(Q_n - C_n A_n), \\
\Gamma_n^+ = \Gamma_n + \mu(A_n - X_{(n)}), \\
\Phi_n^+ = \Phi_n + \mu(Z_{n(n)} - L_n D_n R_n),
\end{cases}
\]

(41)

and \(\mu^+ = \rho \mu\), where \(\rho \geq 1\). The pseudo-code corresponding to our proposed algorithm for TLNMTV is displayed in the table, i.e., Algorithm 3.

### 3.5. Computational Complexity Analysis of Corresponding Algorithms

For any input tensor \(T \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}\) of size \(N\)-order, the entire computational complexity of Algorithm 2 includes the following five major aspects.

Updating the matrix \(L_n\) needs to give the decomposition of QR [see \[13]-\[14]\], and the complexity of which is \(O(I_n r_n^2)\), and the total computational complexity of these variable matrices \(L_n(n = 1, 2, \ldots, N)\) is \(O(\sum_{n=1}^{N} I_n r_n^2)\).

Likewise, the complexity of calculating \(R_n\) is \(O(\sum_{n=1}^{N} r_n^2 \prod_{i=1, i\neq n}^{N} I_i)\).

When updating matrix \(D_n\), it is necessary to perform the product operation between two \(r_n \times r_n\) matrices for solving the LNMS problem, thus the computational complexity of all these variable matrices \(D_n(n = 1, 2, \ldots, N)\) is \(O(\sum_{n=1}^{N} r_n^2)\).
Algorithm 3 TLNMTV

**Input:** an arbitrary incomplete tensor $\mathcal{T}$, the set of indices of the known elements $\Omega$, convergence stop criteria $\epsilon$, maximum number of iteration steps $K$.

**Initialization:** $\mathcal{X}^0 = \mathcal{T}_\Omega$, $\{Z_n^0 = \mathcal{X}^0\}_{n=1}^N$, $\mu^0 > 0$, $\rho > 1$, $\{\alpha_n, r_n, L_n^0, D_n^0, R_n^0\}_{n=1}^N$.

1: while not converged and $k < K$ do
2:   for $n=1:N$ do
3:     Calculate $Q_n^k$ via (32);
4:     Calculate $L_n^k$ via (13-14);
5:     Calculate $R_n^k$ via (15-16);
6:     Calculate $D_n^k$ via (20);
7:     Calculate $Z_n^k$ via (35);
8:     Calculate $A_n^k$ via (38);
9:   end for
10:  Calculate $\mathcal{X}^k$ via (40);
11:  Calculate the multipliers $G_n$, $\Lambda_n$, $\Phi_n$, and $\Gamma_n$ via (41);
12:  $\mu^{k+1} = \rho \mu^k$, $k = k + 1$;
13:  Check whether the convergence condition $\|\mathcal{X}^{k+1} - \mathcal{X}^k\|_\infty \leq \epsilon$ is satisfied
14: end while
15: return $\mathcal{X}^{k+1}$

**Output:** Completed tensor $\mathcal{X} = \mathcal{X}^{k+1}$

Updating each matrix $\mathcal{M}_n$ requires the product operation of these three matrices $L_n$, $D_n$, and $R_n$ in advance, for which the computational complexity of all the corresponding variable matrices $\mathcal{M}_n(n = 1, 2, \ldots, N)$ is $O(\sum_{n=1}^N (I_n r_n^2 + r \prod_{i=1}^N I_i))$.

In addition, the complexity of updating matrix $\mathcal{X}$ is known as $O(\prod_{n} I_n)$. Thus, by summing the above complexities, all the computational complexity of Algorithm 2 is $O(K(\sum_{n=1}^N (I_n r_n^2 + r \prod_{i=1,i\neq n}^N I_i)) + r^2_n + I_n r_n^2 + r \prod_{i=1}^N I_i) + \prod_{n} I_n))$.

Similarly, the entire computational complexity of Algorithm 3 is mainly composed of the following seven parts.

It is easy to know that the computational complexity required to update the matrix $Q_n$ is $O(\sum_{n=1}^N (I_n - 1) \prod_{n} I_n)$.

The computational complexity required to update these matrices $L_n$, $R_n$, $D_n$ is the same as in Algorithm 2.

Updating $Z_n$ requires calculating the product of the three matrices $L_nD_nR_n$ in advance, the complexity of these variable matrices $Z_n(n = 1, 2, \ldots, N)$ is $O(\sum_{n=1}^N (I_n r_n^2 + r \prod_{i=1}^N I_i))$. 

13
Updating $A_n$ requires calculating the inverse of a matrix of order $I_n \times I_n$ and the product matrix of any matrix, the complexity of which is $O(I_n^3 + I_n^2 \prod_{i=1,i\neq n}^N I_i)$. The complexity of these variable matrices $A_n(n = 1, 2, \ldots, N)$ is $O(\sum_{n=1}^N I_n^3 + I_n^2 \prod_{i=1,i\neq n}^N I_i)$.

The computational complexity required to update a tensor $\mathcal{X}$ is $O(\prod_n I_n)$. Thus, the total computational complexity of our proposed Algorithm 3 is $O(K(\sum_{n=1}^N (I_n - 1) \prod_n I_n + I_n r_n^2 + r_n^2 \prod_{i=1,i\neq n}^N I_i + r_n^2 + I_n r_n^2 + r \prod_{i=1}^N I_i + I_n^3 + I_n^2 \prod_{i=1,i\neq n}^N I_i + \prod_n I_n))$.

4. Experiments

In this section, two typical tensor data, i.e., HSI data and MRI data are employed to illustrate the performance of the proposed method. Four quantitative picture quality indices (PQIs) are used to evaluate the quality of recovery, including peak signal–to–noise ratio (PSNR), structural similarity (SSIM) [39], feature similarity (FSIM) [40], and erreur relative globale adimensionnelle de synthèse (ERGAS) [41]. All tests are implemented on the Windows 10 platform and MATLAB (R2019a) with an Intel Core i7-10875H 2.30 GHz and 32 GB of RAM.

We compare our results with nine recently developed state-of-the-art LRTC methods, including the tensor trace norm-based LRTC (HaLRTC) [24], the t-SVD-based TC method [42], alternating direction multiplier method(ALM)-based matrix completion (MC-ALM) [43], the nonconvex tensor rank constraint-based, i.e., the minimax concave plus penalty-based TC (McpTC) and the smoothly clipped absolute deviation penalty-based TC (ScadTC) method [44], the parallel matrix factorization-based LRTC method (TMac) [30], the LRTC with TV on tensor unfolding (LRTC-TV) [31], Tensor Completion based Framelet Representation of Tensor Nuclear Norm(FTNN) [45], and the joint trace/TV-based TC method (Trace/TV) [46]. All comparison methods use the corresponding optimal parameters for the experiment.

| SR | index | Observed | MC-ALM | HaLRTC | TMac | LRTC-TV | Trace-TV | t-SVD | McpTC | ScadTC | FTNN | TLNM | TLNMTV |
|----|-------|---------|--------|--------|------|---------|---------|------|-------|--------|------|------|--------|
| 5% | PSNR | 14.169 | 23.438 | 22.995 | 14.212 | 24.257 | 18.660 | 27.680 | 28.739 | 28.560 | 27.237 | 31.136 | 32.666 |
|    | SSIM | 0.203 | 0.593 | 0.650 | 0.226 | 0.709 | 0.621 | 0.727 | 0.725 | 0.701 | 0.763 | 0.790 | 0.857 |
|    | FSIM | 0.644 | 0.797 | 0.777 | 0.658 | 0.796 | 0.775 | 0.898 | 0.888 | 0.886 | 0.882 | 0.936 | 0.959 |
|    | ERGAS | 910.569 | 332.274 | 350.889 | 906.147 | 308.510 | 562.121 | 210.364 | 195.615 | 199.534 | 221.914 | 139.178 | 116.057 |
| 10% | PSNR | 14.403 | 25.976 | 26.303 | 14.486 | 27.901 | 26.580 | 31.374 | 31.906 | 31.881 | 31.614 | 33.787 | 35.632 |
|    | SSIM | 0.238 | 0.694 | 0.745 | 0.269 | 0.817 | 0.812 | 0.841 | 0.823 | 0.811 | 0.876 | 0.858 | 0.919 |
|    | FSIM | 0.688 | 0.861 | 0.856 | 0.696 | 0.904 | 0.919 | 0.949 | 0.937 | 0.935 | 0.950 | 0.964 | 0.979 |
|    | ERGAS | 886.354 | 252.090 | 246.122 | 878.174 | 204.648 | 247.447 | 139.290 | 134.702 | 139.534 | 143.651 | 135.123 | 85.385 |
| 20% | PSNR | 14.915 | 29.336 | 30.575 | 15.362 | 35.053 | 35.053 | 36.054 | 35.726 | 35.705 | 37.290 | 36.500 | 38.578 |
|    | SSIM | 0.303 | 0.811 | 0.859 | 0.348 | 0.943 | 0.945 | 0.927 | 0.911 | 0.968 | 0.955 | 0.916 | 0.952 |
|    | FSIM | 0.734 | 0.924 | 0.906 | 0.744 | 0.984 | 0.984 | 0.980 | 0.973 | 0.971 | 0.986 | 0.983 | 0.989 |
|    | ERGAS | 835.570 | 173.900 | 152.819 | 805.944 | 93.710 | 93.710 | 83.575 | 86.751 | 89.127 | 71.704 | 78.446 | 62.725 |
Figure 1: (a) Original images. (b) Corresponding sampled images with SR 5%. (c)-(l) and (m) Completed images achieved by nine competing methods and proposed TLNM method and TLNMTV method, respectively. (a) Original image. (b) Observed image. (c) MC-ALM. (d) HaLRTC. (e) TMac. (f) LRTC-TV. (g) Trace/TV. (h) t-SVD. (i) McpTC. (j) ScadTC. (k) FTNN. (l) TLNM. (m) TLNMTV.

Figure 2: (a) Original images. (b) Corresponding sampled images with SR 10%. (c)-(l) and (m) Completed images achieved by nine competing methods and proposed TLNM method and TLNMTV method, respectively. (a) Original image. (b) Observed image. (c) MC-ALM. (d) HaLRTC. (e) TMac. (f) LRTC-TV. (g) Trace/TV. (h) t-SVD. (i) McpTC. (j) ScadTC. (k) FTNN. (l) TLNM. (m) TLNMTV.

4.1. HSI Completion

The HSI test data is used in the experiment came from the open source CAVE data sets (respectively named "chart_and_stuffed_toy", "cloth", "feathers", "flowers", "beads", "oil_painting"). The size of HSI is $512 \times 512 \times 31$, indicating that the spatial resolution is $512 \times 512$ and the spectral resolution is 31, respectively. We show the visual effects of HSI recovery with sampling rates of 5%, 10% and 20%.

\footnote{http://www.cs.columbia.edu/CAVE/databases/multispectral/}
It is not difficult to find that the proposed TLNM method is far superior to the classical HaLRTC method based on SNN and t-SVD method based on TNN. This shows that the proposed SNN method combined with QR decomposition and $L_{2,1}$ norm can improve the recovery effect. At 5% sampling rate, the PSNR value of TLNMTV method is 3dB higher than that of suboptimal method. Even at

| Rate | PSNR Value of TLNMTV Method | PSNR Value of Suboptimal Method |
|------|-----------------------------|---------------------------------|
| 5%   | 2.5db                       | 2.2db                           |
| 10%  | 3.0db                       | 2.7db                           |
| 15%  | 3.5db                       | 3.2db                           |

Table 1: Average quantitative numerical results of HSI at three sampling rates.
Figure 5: The PSNR, SSIM, and FSIM of the results by different methods on all the HSI data with the sampling rate 2.5%.

20% sampling rate, the TLNMTV method is still 1dB higher than the suboptimal FTNN method.
Through the above experimental study, it is found that our method has a better effect at a lower sampling rate. In view of this, we have done experiments at a sampling rate of 2.5%. Fig. 4 shows the visual effect at a 2.5% sampling rate, and Fig. 5 further illustrates the PSNR, SSIM, and FSIM of all methods on all HSIs. It can be seen from Fig. 4 that our method can still obtain more significant recovery effect at a very low sampling rate of 2.5%, which is quite effective in extracting both global visual and structural information and local detail information. And Fig. 5 shows that when the sampling rate is 2.5%, the results obtained by our method are better than those obtained by other state-of-the-art methods.

![Figure 6: (a) Original images. (b) Corresponding sampled images with SR 5% to 30%. (c)-(l) and (m) Completed images achieved by nine competing methods and proposed TLNM method and TLNMTV method, respectively. (a) Original image. (b) Observed image. (c) MC-ALM. (d) HaLRTC. (e) TMac. (f) LRTC-TV. (g) Trace/TV. (h) t-SVD. (i) McpTC. (j) ScadTC. (k) FTNN. (l) TLNM. (m) TLNMTV.](image)

Table 2: Quantitative evaluation of the results for MRI with different SRs.

| SR  | index | Observed | MC-ALM | HaLRTC | TMac | LRTC-TV | Trace-TV | t-SVD | McpTC | ScadTC | FTNN | TLNM | TLNMTV |
|-----|-------|----------|--------|--------|------|---------|----------|-------|-------|--------|------|------|--------|
| 5%  | PSNR  | 9.908    | 17.513 | 16.934 | 9.953| 19.000  | 18.065   | 22.720| 27.520| 27.531 | 24.164| 27.227| 27.541 |
|     | SSIM  | 0.174    | 0.289  | 0.299  | 0.103| 0.530   | 0.464    | 0.514 | 0.746 | 0.748  | 0.653 | 0.724 | 0.782  |
|     | FSIM  | 0.457    | 0.670  | 0.614  | 0.563| 0.663   | 0.657    | 0.770 | 0.855 | 0.851  | 0.819 | 0.845 | 0.817  |
|     | ERGAS | 1025.997 | 431.950| 463.996| 1021.063| 363.138| 414.979 | 242.057| 135.859| 135.675| 203.091| 141.034| 135.572|
| 10% | PSNR  | 10.143   | 20.171 | 19.910 | 10.284| 22.579  | 21.763   | 25.431| 150.154| 150.156| 30.310 | 30.100 | 31.797 |
|     | SSIM  | 0.188    | 0.452  | 0.448  | 0.093| 0.712   | 0.637    | 0.656 | 0.830 | 0.830  | 0.780 | 0.846 | 0.914  |
|     | FSIM  | 0.492    | 0.746  | 0.717  | 0.564| 0.787   | 0.777    | 0.830 | 0.895 | 0.895  | 0.874 | 0.897 | 0.927  |
|     | ERGAS | 998.616  | 317.697| 328.310| 982.744| 240.306| 270.797  | 178.582| 100.559| 100.528| 146.333| 99.056 | 83.339 |
| 20% | PSNR  | 10.656   | 23.458 | 24.151 | 11.143| 27.659  | 26.219   | 29.224| 34.587| 34.448 | 30.880| 34.130| 35.778 |
|     | SSIM  | 0.219    | 0.640  | 0.671  | 0.112| 0.874   | 0.814    | 0.809 | 0.937 | 0.932  | 0.892 | 0.906 | 0.965  |
|     | FSIM  | 0.544    | 0.824  | 0.828  | 0.568| 0.899   | 0.882    | 0.889 | 0.953 | 0.950  | 0.930 | 0.946 | 0.967  |
|     | ERGAS | 941.393  | 216.791| 200.759| 890.839| 134.099| 162.074  | 116.642| 60.244 | 61.197 | 53.971 | 63.738 | 52.713 |
| 30% | PSNR  | 11.231   | 25.921 | 27.584 | 11.950| 30.867  | 29.330   | 32.172| 37.987| 37.206 | 33.712 | 37.591 | 38.364 |
|     | SSIM  | 0.253    | 0.760  | 0.814  | 0.144| 0.932   | 0.895    | 0.888 | 0.977 | 0.966  | 0.940 | 0.972 | 0.980  |
|     | FSIM  | 0.575    | 0.875  | 0.895  | 0.580| 0.944   | 0.930    | 0.938 | 0.979 | 0.974  | 0.958 | 0.975 | 0.984  |
|     | ERGAS | 881.074  | 163.086| 134.957| 812.918| 92.514  | 113.377  | 83.479| 40.760| 44.327 | 67.186 | 42.717 | 39.060 |
4.2. MRI Completion

We test the performance of the proposed method and the comparison method on MRI data with the size of 181 × 217 × 121. Fig. 6 shows the visual effects of MRI images at the sampling rate of 5%, 10%, 20% and 30% respectively. It can be found from the figure that for the visual effect of MRI image recovery, our method is superior to the comparison method in both global and local information recovery. Further, Fig. 7 describes the specific values of PSNR, SSIM and FSIM of each slice at the sampling rate of 20% and 30%. As one can see from Fig. 7 that the TLNMTV method is the optimal

---

Figure 7: PSNR, SSIM, and FSIM values comparison of different methods for each slice on MRI data sets, (a), (c), (e) under SR 20%, (b), (d), (f) under SR 30%.

\[\text{http://brainweb.bic.mni.mcgill.ca/brainweb/selection_normal.html}\]
method. Table 2 reports the PSNR, SSIM, FSIM, ERGAS values for each sample rate. It can be seen from the table that at different sampling rates, the TLNM method is obviously superior to the HaLRTC and t-SVD, and the results obtained are very close to those obtained by non-convex method McpTC. Both methods are based on SNN, which indicates that the convex TLNM method is very efficient. In addition, the TLNMTV method is obviously superior to the suboptimal method McpTC. Even when the sampling rate is 20%, the PSNR value of TLNMTV method is 1.2 dB higher than that of McpTC.

5. Parameters setting and Convergency Behaviours

We obtain the optimal solutions of the two algorithm parameters through experimental tests. Table 3 lists the optimal parameters of \( r_n \) for different images with different sampling rates. We let \( \alpha \) and \( \beta = [1/N, 1/N, \ldots, 1/N] \). In order to observe the convergence behavior of the algorithm, the relative error (RE) of the recovery tensor is defined as follows:

\[
    RE := \frac{\|X_{rec} - X_o\|_F}{\|X_o\|_F},
\]

(42)

where \( X_{rec} \) is recovery tensor and \( X_o \) is origin tensor. Fig. 8 shows the iterative RE of the proposed algorithm for HSI and MRI. As the iteration progresses, the RE can become smaller, which guarantees the convergence of the proposed algorithm.

| Method       | TLNM       | TLNMTV     |
|--------------|------------|------------|
| SR           | 2.50%      | 5%         | 10%        | 20%  | 2.50% | 5% | 10% | 20% |
| chart and stuffed toy | 50 50 4 | 85 85 4 | 125 125 4 | 205 205 4 | 60 60 4 | 95 95 4 | 145 145 4 | 205 205 4 |
| cloth        | 50 50 4 | 85 85 4 | 125 125 4 | 205 205 4 | 130 130 4 | 160 160 4 | 235 235 4 | 275 275 4 |
| feather      | 40 40 4 | 65 65 4 | 105 105 4 | 205 205 4 | 50 50 4 | 95 95 4 | 145 145 4 | 205 205 4 |
| flower       | 50 50 4 | 75 75 4 | 125 125 4 | 205 205 4 | 50 50 4 | 95 95 4 | 165 165 4 | 265 265 4 |
| beads        | 50 50 4 | 115 115 4 | 175 175 4 | 255 255 4 | 110 110 4 | 155 155 4 | 215 215 4 | 235 235 4 |
| oil painting | 60 60 4 | 95 95 4 | 185 185 4 | 235 235 4 | 80 80 4 | 135 135 4 | 205 205 4 | 250 250 4 |
| SR           | 5%        | 10%       | 20%        | 30%  | 5%   | 10% | 20% | 30% |
| MRI          | 50 60 30 | 60 70 40 | 70 90 55 | 90 110 60 | 50 60 30 | 80 90 40 | 100 120 55 | 110 130 60 |
Figure 8: The convergence behaviours of LRTC Algorithm, with respect to different sampling rates.
6. Conclusion

This paper integrates SNN with QR decomposition and $L_{2,1}$ norm to solve the LRTC problem, and proposes the TLNM method. The TV regularization term is further introduced to improve the local prior information and make the recovery effect of the TLNM method more ideal. HSI and MRI are tested at different sampling rates. Experiments show that our TLNM method is more efficient than the TNN-based TLNM-TQR method and outperforms the classic HaLRTC and t-SVD methods. Besides, the TLMNTV method surpasses other compared methods in this paper, thus confirming the high efficiency of our proposed method. It is particularly worth emphasizing that our method still achieves fairly good HSI recovery when the SR is extremely low, i.e. 2.5%. In future work, we will try to introduce other more efficient forms, such as non-convex transformations to replace the nuclear norm, to further improve the recovery performance of our method.

References

[1] Y.-M. Huang, H.-Y. Yan, Y.-W. Wen, X. Yang, Rank minimization with applications to image noise removal, Information Sciences 429 (2018) 147–163.

[2] B. Madathil, S. N. George, Twist tensor total variation regularized-reweighted nuclear norm based tensor completion for video missing area recovery, Information Sciences 423 (2018) 376–397.

[3] H. Zhang, X. Liu, H. Fan, Y. Li, Y. Ye, Tensor full feature measure and its nonconvex relaxation applications to tensor recovery, arXiv preprint arXiv:2109.12257 (2021).

[4] J.-H. Yang, X.-L. Zhao, T.-Y. Ji, T.-H. Ma, T.-Z. Huang, Low-rank tensor train for tensor robust principal component analysis, Applied Mathematics and Computation 367 (2020) 124783.

[5] J. Xue, Y. Zhao, W. Liao, J. Cheung-Wai Chan, Nonconvex tensor rank minimization and its applications to tensor recovery, Information Sciences 503 (2019) 109–128. doi:https://doi.org/10.1016/j.ins.2019.06.061

[6] X. Fu, W.-K. Ma, J. M. Bioncas-Dias, T.-H. Chan, Semiblind hyperspectral unmixing in the presence of spectral library mismatches, IEEE Transactions on Geoscience and Remote Sensing 54 (9) (2016) 5171–5184.

[7] J. Xue, Y. Zhao, W. Liao, J. C.-W. Chan, Nonlocal low-rank regularized tensor decomposition for hyperspectral image denoising, IEEE Transactions on Geoscience and Remote Sensing 57 (7) (2019) 5174–5189. doi:10.1109/TGRS.2019.2897316
[8] L. Zhang, L. Zhang, D. Tao, X. Huang, Tensor discriminative locality alignment for hyperspectral image spectral–spatial feature extraction, IEEE Transactions on Geoscience and Remote Sensing 51 (1) (2013) 242–256. doi:10.1109/TGRS.2012.2197860

[9] Q. Li, D. Schonfeld, Multilinear discriminant analysis for higher-order tensor data classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (12) (2014) 2524–2537. doi:10.1109/TPAMI.2014.2342214

[10] X. Cao, X. Wei, Y. Han, D. Lin, Robust face clustering via tensor decomposition, IEEE Transactions on Cybernetics 45 (11) (2015) 2546–2557. doi:10.1109/TCYB.2014.2376938

[11] J. Xue, Y. Zhao, W. Liao, J. C.-W. Chan, S. G. Kong, Enhanced sparsity prior model for low-rank tensor completion, IEEE Transactions on Neural Networks and Learning Systems 31 (11) (2020) 4567–4581. doi:10.1109/TNNLS.2019.2956153

[12] E. Acar, D. M. Dunlavy, T. G. Kolda, M. Mørup, Scalable tensor factorizations for incomplete data, Chemometrics and Intelligent Laboratory Systems 106 (1) (2011) 41–56.

[13] P. Tichavský, A.-H. Phan, A. Cichocki, Numerical cp decomposition of some difficult tensors, Journal of Computational and Applied Mathematics 317 (2017) 362–370.

[14] R. Harshman, Foundations of the parafac procedure: Model and conditions for an“explanatory”multi-mode factor analysis, UCLA Work. Pap. Phon. 16 (11 1969).

[15] L. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (1966) 279–311.

[16] W. Cao, Y. Wang, J. Sun, D. Meng, C. Yang, A. Cichocki, Z. Xu, Total variation regularized tensor rpca for background subtraction from compressive measurements, IEEE Transactions on Image Processing 25 (9) (2016) 4075–4090.

[17] M. E. Kilmer, C. D. Martin, Factorization strategies for third-order tensors, Linear Algebra and its Applications 435 (3) (2011) 641–658.

[18] Z. Zhang, G. Ely, S. Aeron, N. Hao, M. Kilmer, Novel methods for multilinear data completion and de-noising based on tensor-svd, 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014) 3842–3849.

[19] C. J. Hillar, L.-H. Lim, Most tensor problems are np-hard, Journal of the ACM (JACM) 60 (6) (2013) 1–39.
[20] Y.-F. Li, K. Shang, Z.-H. Huang, Low tucker rank tensor recovery via admm based on exact and inexact iteratively reweighted algorithms, Journal of Computational and Applied Mathematics 331 (2018) 64–81.

[21] X. Li, M. K. Ng, G. Cong, Y. Ye, Q. Wu, Mr-ntd: Manifold regularization nonnegative tucker decomposition for tensor data dimension reduction and representation, IEEE Transactions on Neural Networks and Learning Systems 28 (8) (2017) 1787–1800. doi:10.1109/TNNLS.2016.2545400.

[22] K. Braman, Third-order tensors as linear operators on a space of matrices, Linear Algebra and its Applications 433 (7) (2010) 1241–1253.

[23] M. E. Kilmer, C. D. Martin, Factorization strategies for third-order tensors, Linear Algebra and its Applications 435 (3) (2011) 641–658.

[24] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2013) 208–220. doi:10.1109/TPAMI.2012.39.

[25] N. Paragios, Y. Chen, O. Faugeras, Handbook of Mathematical Models in Computer Vision —— Total Variation Image Restoration: Overview and Recent Developments, Vol. 10.1007/0-387-28831-7, Springer Science & Business Media, 2006.

[26] X. Cao, L. Yang, X. Guo, Total variation regularized rpca for irregularly moving object detection under dynamic background, IEEE Transactions on Cybernetics 46 (4) (2016) 1014–1027. doi:10.1109/TCYB.2015.2419737.

[27] J. Xue, Y. Zhao, W. Liao, J. Cheung-Wai Chan, Total variation and rank-1 constraint rpca for background subtraction, IEEE Access 6 (2018) 49955–49966. doi:10.1109/ACCESS.2018.286831.

[28] B. Madathil, S. N. George, Twist tensor total variation regularized-rewighted nuclear norm based tensor completion for video missing area recovery, Information Sciences 423 (2018) 376–397.

[29] H. Wang, F. Nie, H. Huang, Low-rank tensor completion with spatio-temporal consistency, Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence 4 (2014) 2846–2852.

[30] Y. Xu, R. Hao, W. Yin, Z. Su, Parallel matrix factorization for low-rank tensor completion, Inverse Problems and Imaging 9 (2015) 601–624. doi:10.3934/ipi.2015.9.601.

[31] X. Li, Y. Ye, X. Xu, Low-rank tensor completion with total variation for visual data inpainting, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (2017) 2210–2216.
[32] X. Guo, Y. Ma, Generalized tensor total variation minimization for visual data recovery?, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 3603–3611.

[33] Q. Liu, F. Davoine, J. Yang, Y. Cui, Z. Jin, F. Han, A fast and accurate matrix completion method based on qr decomposition and $l_{2,1}$-norm minimization, IEEE Transactions on Neural Networks and Learning Systems 30 (3) (2019) 803–817.

[34] Y. Zheng, A.-B. Xu, Tensor completion via tensor qr decomposition and $l_{2,1}$-norm minimization, Signal Processing 189 (2021) 108240.

[35] S. Boyd, N. Parikh, E. Chu, Distributed optimization and statistical learning via the alternating direction method of multipliers, Now Publishers Inc, 2011.

[36] Z. Lin, R. Liu, Z. Su, Linearized alternating direction method with adaptive penalty for low-rank representation, Advances in Neural Information Processing Systems 24 (2011) 612–620.

[37] T. G. Kolda, B. W. Bader, Tensor decompositions and applications, SIAM review 51 (3) (2009) 455–500.

[38] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, ICML 2010 - Proceedings, 27th International Conference on Machine Learning (2010) 663–670.

[39] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612. doi:10.1109/TIP.2003.819861

[40] L. Zhang, L. Zhang, X. Mou, D. Zhang, Fsim: A feature similarity index for image quality assessment, IEEE Transactions on Image Processing 20 (8) (2011) 2378–2386. doi:10.1109/TIP.2011.2109730

[41] L. Wald, Data fusion: definitions and architectures: fusion of images of different spatial resolutions, Presses des MINES, 2002.

[42] Z. Zhang, S. Aeron, Exact tensor completion using t-svd, IEEE Transactions on Signal Processing 65 (6) (2017) 1511–1526. doi:10.1109/TSP.2016.2639466

[43] Z. Lin, M. Chen, Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, Mathematical Programming 9 (09 2010).

[44] W. Cao, Y. Wang, C. Yang, X. Chang, Z. Han, Z. Xu, Folded-concave penalization approaches to tensor completion, Neurocomputing 152 (2015) 261–273.
[45] T.-X. Jiang, M. K. Ng, X.-L. Zhao, T.-Z. Huang, Framelet representation of tensor nuclear norm for third-order tensor completion, IEEE Transactions on Image Processing 29 (2020) 7233–7244. doi:10.1109/TIP.2020.3000349

[46] M. M, H. LK, A. SM, Algorithms for sparse nonnegative tucker decompositions, Neural computation 20 (8) (2008) 2112–2131.