Journal of Scientific Computing (2025) 102:68
https://doi.org/10.1007/s10915-025-02792-6

Data-Driven Tensor Dictionary Learning for Image Alignment

Quan Yu[1] · **Minru Bai**[1]

Received: 2 January 2024 / Revised: 30 November 2024 / Accepted: 6 January 2025

Abstract
Image alignment is an important problem in computer vision, which can be solved by tensor based methods that are robust to noise and have satisfactory performance. However, these methods face two common challenges: (1) they have high computational cost when dealing with large-scale tensor data; (2) they ignore the local structures within and across images. To overcome these challenges, we propose an efficient data-driven tensor dictionary learning (DTDL) model for image alignment. In our DTDL model, we factorize the underlying third order tensor into a coefficients tensor and three dictionary matrices of smaller sizes, which reduces the dimensionality and complexity of the problem. We also exploit the generalized hyper-Laplacian regularization to preserve the local structures that are embedded in the underlying tensor and represented by the dictionary framework. Furthermore, we prove that our proximal linearized alternating direction method of multipliers algorithm can generate a sequence that converges to a Karush–Kuhn–Tucker point under very mild conditions. We conduct experiments on image alignment and face recognition tasks, and show that our method outperforms state-of-the-art methods in terms of performance and efficiency.

Keywords Image alignment · Dictionary learning · Data-driven · Hyper-Laplacian regularization

1 Introduction

Image alignment has gained increasing attention in face recognition [41] and machine learning communities [18]. Based on the existing literature, image alignment methods can be broadly classified into two categories: congealing based methods and low rank based methods.

Congealing based methods: Congealing is a nonparametric technique for separating a set of images into sets of approximately independent "ingredients" [10]. It can be applied to image alignment problems. For instance, Learned-Miller [10] proposed a congealing method that minimizes the sum of the pixel-stack entropies with image transformations. Cox et al. [3, 4] proposed a least squares congealing method that uses a different alignment measure

✉ Minru Bai
minru-bai@hnu.edu.cn

Quan Yu
quanyu@hnu.edu.cn

[1] School of Mathematics, Hunan University, Changsha 410082, Hunan, China

Published online: 24 January 2025

than entropy and minimizes it for batch image alignment tasks. However, these methods are ineffective for aligning linearly correlated images that suffer from common real world degradation, such as large illumination variations and gross pixel corruptions or partial occlusions [16].

Low rank based methods: Low rank matrix based methods, which are shown to be more promising than congealing based methods [16], decompose the transformed images into a low rank matrix of aligned images and a sparse matrix of errors, and seek the optimal domain transformations while minimizing the rank and sparsity. A unified model of these methods can be mathematically expressed as

$$\min_{X,E,\Gamma} \ \operatorname{rank}(C) + \lambda \|E\|_0, \quad \text{s.t.} \quad Y \circ \Gamma = X + E. \tag{1}$$

Here the ℓ_0-norm $\|\cdot\|_0$ counts the number of nonzero entries. $Y = [\operatorname{vec}(I_1), \ldots, \operatorname{vec}(I_{n_3})] \in \mathbb{R}^{n_1 n_2 \times 3}$ is the data matrix of n_3 input images $I_k \in \mathbb{R}^{n_1 \times n_2}$. $\Gamma = \{\tau_1, \ldots, \tau_{I_3}\}$ are the transformations. $Y \circ \Gamma$ means that the transformation τ_k is applied to the image I_k for $k = 1, 2, \ldots, n_3$ [16]. To solve (1), RASL [16] applies the convex relaxation theory to $\operatorname{rank}(\cdot)$ and $\|\cdot\|_0$ to obtain a new optimization problem:

$$\min_{X,E,\Gamma} \ \|X\|_* + \lambda \|E\|_1, \quad \text{s.t.} \quad Y \circ \Gamma = X + E, \tag{2}$$

where $\|\cdot\|_*$ denotes the nuclear norm, which is the sum of the singular values of a matrix, and $\|\cdot\|_1$ denotes the ℓ_1-norm, which is the sum of the absolute values of the matrix entries. To reduce the high computational cost of SVD in each iteration of numerical methods for (2), He et al. [6] proposed a matrix factorization method that maintains the low rank structure of a matrix. This leads to a reformulation of (1) as

$$\min_{U,W,E,\Gamma} \ \|E\|_1, \quad \text{s.t.} \quad Y \circ \Gamma = UW + E. \tag{3}$$

Low rank matrix based methods capture the low rankness across the images, yet destroys the intrinsic structure of individual images. To alleviate this limitation, Zhang et al. [41], Qiu et al. [21], and Xia et al. [30] consider low rank tenor based methods, which performs more naturally and auspiciously than vector linear representation for image data analysis. This leads to a reformulation of (1) as

$$\min_{\mathcal{X},\mathcal{E},\Gamma} \ \operatorname{rank}(\mathcal{X}) + \lambda \|\mathcal{E}\|_0, \quad \text{s.t.} \quad \mathcal{Y} \circ \Gamma = \mathcal{X} + \mathcal{E}. \tag{4}$$

Here $\mathcal{Y} \circ \Gamma$ represents that the transformation τ_k is applied to each frontal slice $\mathcal{Y}(:, :, k)$ for $k = 1, 2, \ldots, n_3$.

Although the existing tensor based methods have achieved satisfactory performance, they still suffer from the following limitations: (1) they are computationally expensive since they require calculating the singular value decompositions (SVDs) of many large-scale matrices or performing numerous matrix multiplications; (2) they often ignore the locality and similarity information of each image and between images. To address these two issues, we propose a novel image alignment method using data-driven tensor dictionary learning (DTDL) model. Similar to ℓ_p+ADMM [41], NCALTS [21] and TFM-TTP [30], DTDL is also a tensor based method. But instead of using the rank of $\mathcal{X}$, we factorize the low rank tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ into the product of one coefficients tensor $\mathcal{L} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ and three dictionary matrices $D_i \in \mathbb{R}^{n_i \times r_i}$ of smaller sizes, where r_i is usually much smaller than n_i, especially r_3. DTDL model based on this factorization has a performance guarantee, since by Theorems 1 and 2, DTDL and dictionary-free model based on Tucker rank [41] and tubal rank [21] are equivalent. Unlike ℓ_p+ADMM and NCALTS, which require calculating the singular

value decompositions (SVDs) of many large-scale matrices (specifically, n_3 matrices of size $n_1 \times n_2$), and TFM-TTP, which involves extensive matrix multiplications due to the large third dimension n_3, DTDL only computes the SVDs of a few small-scale matrices (specifically, r_3 matrices of size $r_1 \times r_2$). This results in a lower computational cost per iteration, $\mathcal{O}(n_1 n_2 n_3 \sum_{i=1}^{3} r_i)$, compared to the computational complexities of ℓ_p+ADMM, NCALTS, and TFM-TTP, which are $\mathcal{O}(n_1 n_2 n_3 \sum_{i=1}^{3} n_i)$, $\mathcal{O}(n_1 n_2 n_3(\min\{n_1, n_2\}+n_3))$, and $\mathcal{O}(n_1 n_2 n_3(\hat{r}+n_3))$, respectively, where $\hat{r}$ is the estimated tensor tubal rank of $\mathcal{X}$. Moreover, DTDL can preserve the local structures of $\mathcal{X}$ using adaptively learned dictionary matrices based on the locality and similarity information. Therefore, the main purpose of this paper is to propose an efficient image alignment method which integrates the local structures in a dictionary framework.

To summarize, this paper makes the following contributions:

(i) We propose an efficient data-driven tensor dictionary learning (DTDL) model for image alignment, which factorizes the underlying third order tensor into a coefficients tensor and three dictionary matrices of smaller sizes, reducing the dimensionality and complexity of the problem.
(ii) We establish the equivalence between the proposed model and the dictionary-free model based on the Tucker rank and the tubal rank, respectively. These theoretical results are rarely provided by previous works in this field.
(iii) We use the generalized hyper-Laplacian regularization on the dictionary atoms, which allows us to adapt the dictionary to the local structures of the data tensor, instead of just enforcing orthogonality or normalization constraints on it. This way, we preserve the local structure information of the data while we update the dictionary.
(iv) We prove that the sequence generated by the proposed proximal linearized alternating direction method of multipliers algorithm can converge to a Karush–Kuhn–Tucker point under very mild conditions. Experimental results show noticeable improvement over the state-of-the-art image alignment methods in terms of both alignment accuracy and computational efficiency on various datasets.

This paper is organized as follows. In Sect. 2, we introduce some notation and preliminaries on Tucker decomposition, tensor singular value decomposition, tensor norm and its proximal mapping. In Sect. 3, we present the proposed method for image alignment. In Sect. 4, we provide an algorithm with convergence analysis for solving the proposed model. In Sect. 5, we report numerical examples to demonstrate the effectiveness and efficiency of the proposed method. In Sect. 6, we conclude the paper.

2 Preliminary Knowledge on Tensor

Before proceeding, we first present some notations here. For a positive integer n, $[n] := \{1, 2, \ldots, n\}$. Scalars, vectors, matrices and tensors are denoted as lowercase letters, boldface lowercase letters, uppercase letters and calligraphic letters, respectively, e.g., x, $\boldsymbol{x}$, X, $\mathcal{X}$. For a third order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its (i, j, k) entry is denoted by $\mathcal{X}_{ijk}$ or $\mathcal{X}(i, j, k)$, and we use the notations $X^{(k)}$ to denote its kth frontal slice. The inner product of two third order tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \mathcal{X}_{ijk} \mathcal{Y}_{ijk}$. The Frobenius norm and ℓ_p-norm with $p \in (0, 2)$ of $\mathcal{X}$ are $\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$ and $\|\mathcal{X}\|_p = (\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} |\mathcal{X}_{ijk}|^p)^{1/p}$, respectively. Let $\sigma_s(X)$ denote the sth largest singular value of X. The spectral norm of matrix X is defined as $\|X\| = \sqrt{\sigma_1(X^T X)}$.

Before proceeding with the model, we overview some tensor related concepts.

2.1 Tucker Decomposition

In this part, we review the Tucker decomposition and introduce some related tensor operations and properties. For more details, we refer the readers to the excellent review paper [9].

Definition 1 *(Tensor mode-i product)* The mode-i product of $\mathcal{X} \in \mathbb{R}^{n_1 \times \ldots \times n_m}$ with $A \in \mathbb{R}^{r_i \times n_i}$ is written as $\mathcal{X} \times_i A \in \mathbb{R}^{n_1 \times \ldots n_{i-1} \times r_i \times n_{i+1} \times \ldots \times n_m}$, defined component-wisely by

$$(\mathcal{X} \times_i A)_{s_1 \ldots s_{i-1} j s_{i+1} \ldots s_m} = \sum_{s_i=1}^{n_i} \mathcal{X}_{s_1 s_2 \ldots s_m} A_{j s_i}.$$

For matrices A, B, C and D of appropriate sizes, there hold

(1) $\mathcal{X} \times_i A \times_j B = (\mathcal{X} \times_i A) \times_j B = \left(\mathcal{X} \times_j B\right) \times_i A$ for $i \neq j$;
(2) $\mathcal{X} \times_i C \times_i D = \mathcal{X} \times_i (DC)$.

Definition 2 *(Mode-i unfolding and folding operations)* The mode-i unfolding $X_{(i)}$ of $\mathcal{X} \in \mathbb{R}^{n_1 \times \ldots \times n_m}$ is a matrix in $\mathbb{R}^{n_i \times \prod_{s \neq i} n_s}$, in which the columns consist of vectors that keep all indices except i. The mode-i folding operation is $\mathcal{X} = \text{fold}_{(i)}(X_{(i)})$.

Let $\mathcal{T} = \mathcal{G} \times_1 V^{(1)} \times_2 V^{(2)} \times \cdots \times_m V^{(m)}$. Then for any $i \in [m]$, one has

$$T_{(i)} = V^{(i)} G_{(i)} \left(V^{(m)} \otimes \ldots \otimes V^{(i+1)} \otimes V^{(i-1)} \otimes \ldots \otimes V^{(1)}\right)^T,$$

where $A \otimes B$ is the Kronecker product of A and B.

Definition 3 *(Orthogonal Tucker decomposition)* Let $\mathcal{T} = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}$ be a third order tensor, where $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$, $r_i = \text{rank}(T_{(i)})$ and $U^{(i)} \in \mathbb{R}^{n_i \times r_i}$ is column orthogonal for all $i \in [3]$. Then $\mathcal{G}$ is called the core tensor and the above equation is called an orthogonal Tucker decomposition of $\mathcal{T}$.

Remark 1 If $U^{(3)}$ is the identity matrix, then the orthogonal Tucker decomposition reduces to the orthogonal Tucker2 decomposition.

2.2 Tensor Singular Value Decomposition

Let $\bar{\mathcal{X}} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ be the result of Discrete Fourier Transformation (DFT) of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ along the 3th mode. Specifically, let $F = [\boldsymbol{f_1}, \ldots, \boldsymbol{f_{n_3}}] \in \mathbb{C}^{n_3 \times n_3}$ with

$$\boldsymbol{f_i} = \left[\omega^{0 \times (i-1)}; \omega^{1 \times (i-1)}; \ldots; \omega^{(n_3-1) \times (i-1)}\right] \in \mathbb{C}^{n_3},$$

$\omega = e^{-\frac{2\pi \mathfrak{b}}{n_3}}$ and $\mathfrak{b} = \sqrt{-1}$. Then $\bar{\mathcal{X}} = \mathcal{X} \times_3 F$, which can be computed by Matlab command "$\bar{\mathcal{X}} = \text{fft}(\mathcal{X}, [\,], 3)$". Furthermore, $\mathcal{X}$ can be computed by $\bar{\mathcal{X}}$ with the inverse DFT $\mathcal{X} = \text{ifft}(\bar{\mathcal{X}}, [\,], 3)$.

Definition 4 *(T-product* [8]*)* The t-product between $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{Y} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is defined as

$$\mathcal{X} * \mathcal{Y} = fold\,(bcirc(\mathcal{X}) \cdot unfold\,(\mathcal{Y})) \in \mathbb{R}^{n_1 \times n_4 \times n_3},$$

where

$$bcirc(\mathcal{X}) = \begin{bmatrix} X^{(1)} & X^{(n_3)} & \cdots & X^{(2)} \\ X^{(2)} & X^{(1)} & \cdots & X^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ X^{(n_3)} & X^{(n_3-1)} & \cdots & X^{(1)} \end{bmatrix},$$

$unfold(\mathcal{Y}) = [Y^{(1)}; Y^{(2)}; \cdots; Y^{(n_3)}] \in \mathbb{R}^{n_2 n_3 \times n_4}$ and its inverse operator fold is defined as $fold(unfold(\mathcal{Y})) = \mathcal{Y}$.

Definition 5 *(F-diagonal tensor* [8]*)* A tensor is called f-diagonal if each of its frontal slices is a diagonal matrix.

Definition 6 *(Conjugate transpose* [8]*)* The conjugate transpose of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the tensor $\mathcal{X}^T \in \mathbb{R}^{n_2 \times n_1 \times n_3}$ obtained by conjugate transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through n_3.

Definition 7 *(Identity tensor* [8]*)* 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 other frontal slices are all zeros.

Definition 8 *(Orthogonal tensor* [8]*)* A tensor $\mathcal{X} \in \mathbb{R}^{n \times n \times n_3}$ is orthogonal if it satisfies $\mathcal{X}^T * \mathcal{X} = \mathcal{X} * \mathcal{X}^T = \mathcal{I}$.

We now introduce a new tensor decomposition framework, t-SVD.

Definition 9 *(T-SVD* [8]*)* For a given tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, it can be factorized as

$$\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T,$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors and $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor.

Definition 10 *(Tensor tubal rank)* [7] For any $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its tubal rank is defined as $\text{rank}_t(\mathcal{X}) = \max_{k \in [n_3]} \text{rank}(\bar{X}^{(k)})$.

2.3 Tensor Norm

To proceed, we need to define the generalized nonconvex low rank and sparse tensor norm, which incorporates some well-known functions such as ℓ_q penalty function, minimax concave penalty function, and so on.

Definition 11 Given $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $n = \min\{n_1, n_2\}$, the generalized nonconvex low rank tensor norm of $\mathcal{X}$ is defined as

$$\|\mathcal{X}\|_{\circledast}^{\psi} = \frac{1}{n_3} \sum_{k=1}^{n_3} \sum_{s=1}^{n} \psi\left(\sigma_s\left(\bar{X}^{(k)}\right)\right). \tag{5}$$

Indeed, when $\psi(x) = x$, $\|\mathcal{X}\|_{\circledast}^{\psi}$ would degrade into tensor nuclear norm (TNN) [23].

Definition 12 Given $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the generalized nonconvex sparse tensor norm of $\mathcal{X}$ is defined as

$$\|\mathcal{X}\|_1^{\psi} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \psi\left(\mathcal{X}_{ijk}\right). \tag{6}$$

The function ψ is nonconvex and satisfies the following assumptions:

Assumption 1 (a) ψ be a proper, lower semicontinuous and y-axis symmetric function;
(b) ψ be a concave and monotonically nondecreasing function on $[0, +\infty)$ with $\psi(0) = 0$.

Example 1 Most existing functions satisfy Assumption 1. Below, we present three specific instances:

(1) ℓ_q penalty: $\psi(x) = |x|^q$, $0 < q < 1$;
(2) Smoothly clipped absolute deviation (SCAD) penalty:

$$\psi(x) = \begin{cases} \xi|x|, & \text{if } |x| < \xi, \\ \frac{2b\xi|x| - x^2 - \xi^2}{2(b-1)}, & \text{if } \xi \leqslant |x| < b\xi, \\ (b+1)\xi^2/2, & \text{if } |x| \geqslant b\xi, \end{cases}$$

where $b > 1$, $\xi > 0$;
(3) Minimax concave penalty (MCP):

$$\psi(x) = \begin{cases} c|x| - \frac{1}{2\eta}x^2, & \text{if } |x| \leqslant c\eta, \\ \frac{c^2\eta}{2}, & \text{if } |x| > c\eta, \end{cases}$$

where $\eta, c > 0$.

3 Proposed DTDL for Image Alignment

In this section, we describe the use of data-driven tensor dictionary learning (DTDL) model for the task of aligning and removing noise from a batch of linearly correlated images. We first present the dictionary learning model for reducing processing time, then incorporate the original tensor spatial and temporal information into the corresponding dictionary to learn it adaptively, and finally, deduce the new image alignment model.

3.1 Tensor Dictionary Learning Model

Due to the high dimensionality of $\mathcal{X}$, existing image alignment algorithms [16, 21, 41] have high computational costs and poor scalability. To overcome these obstacles, we replace $\mathcal{X}$ with $\mathcal{L} \times_1 D_1 \times_2 D_2 \times_3 D_3$ and our tensor dictionary learning model is formulated as

$$\begin{aligned} \min_{\mathcal{L}, D_i, \mathcal{E}, \Gamma} \; & R_1(\mathcal{L}) + \lambda_1 R_2(\mathcal{E}) \\ \text{s.t.} \quad & \mathcal{Y} \circ \Gamma = \mathcal{L} \times_1 D_1 \times_2 D_2 \times_3 D_3 + \mathcal{E}, \end{aligned} \tag{7}$$

where matrices $D_i \in \mathbb{R}^{n_i \times r_i}, i \in [3]$ represents the dictionary corresponding to the ith direction, and $\mathcal{L} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ is the corresponding low rank coding coefficients tensor. The regularization terms $R_1(\mathcal{L})$ and $R_2(\mathcal{E})$ are used to depict the low rank and sparse properties of $\mathcal{L}$ and $\mathcal{E}$, respectively.

Remark 2 When n_i is large (in practice, this often happens when the number of images to be aligned is large, i.e., n_3 is large), we can set $r_i \ll n_i$ to save computation by ignoring unimportant data. When n_i is small, we can set $r_i \approx n_i$ to enhance the alignment effect by keeping the details. Therefore, by controlling the value of r_i, we can both reduce the computation and improve the alignment effect.

Remark 3 Unlike traditional dictionary learning techniques, which enforce the sparsity or tubal sparsity of coefficients [17, 42], we use a specific low rank structure of the coefficients, which allows us to complete $\mathcal{X}$ accurately by combining features linearly, together with the learned dictionary. Please see Subsection 5.3-1) for detailed comparisons of sparsity and low rankness.

Now we are ready to establish the equivalence between (7) and dictionary-free model based on Tucker rank [41] and tubal rank [21], respectively. The proofs of the following two theorems are provided in "Appendix A" and "Appendix B".

Theorem 1 *Problem* (7) *is equivalent to*

$$\min_{\mathcal{X},\mathcal{E},\Gamma} \sum_{i=1}^{3} \operatorname{rank}\left(X_{(i)}\right) + \lambda_1 R_2\left(\mathcal{E}\right), \quad s.t. \quad \mathcal{Y} \circ \Gamma = \mathcal{X} + \mathcal{E}, \tag{8}$$

which is the problem studied in [41], *under the condition* $R_1(\mathcal{L}) = \sum_{i=1}^{3} \operatorname{rank}(L_{(i)})$.

Theorem 2 *Problem* (7) *is equivalent to*

$$\min_{\mathcal{X},\mathcal{E},\Gamma} \operatorname{rank}_t\left(\mathcal{X}\right) + \lambda_1 R_2\left(\mathcal{E}\right), \quad s.t. \quad \mathcal{Y} \circ \Gamma = \mathcal{X} + \mathcal{E}, \tag{9}$$

which is the problem studied in [21], *under the condition* $R_1(\mathcal{L}) = \operatorname{rank}_t(\mathcal{L})$ *and* $D_3 = I$.

Remark 4 From the above two theorems, we can see that our model and the model based on Tucker rank and Tubal rank can obtain the same global optimal solution under certain conditions, but our model reduces the computational cost by decomposing a large tensor data into a product of a small coefficient tensor and three small dictionary matrices.

3.2 Dictionary Learning with Generalized Hyper-Laplacian Regularization

The dictionary is a key component of the model that relies on dictionary learning. However, the existing methods of selecting the dictionary are coarse, and can be divided into two main categories:

(1) learning a fixed dictionary in advance from the original data tensor using principal component analysis (PCA) or other techniques [24, 27];

(2) learning a dynamic dictionary by adding a kernel norm or other regularization term to the objective function to enforce low rank or smoothness constraints on the dictionary [34, 40]; or by adding normalized constraints [19], orthogonal constraints [22] or other constraints to avoid the pathological case or reduce the processing time.

These two methods do not adaptively adjust the dictionary learning method according to the different original data tensor. To address these issues, we propose a prior-based dictionary learning model that incorporates a generalized manifold structure.

To achieve this, we present Theorem 3 as follows:

Theorem 3 *[32] Let* $\mathcal{X} = \mathcal{L} \times_1 D_1 \times_2 D_2 \times_3 D_3$. *Define the operator* ∇ *as a linear gradient operation characterized by* $\nabla_{i,i} = 1$ *and* $\nabla_{i,i+1} = -1$, *with all other entries being zero. Then, we have*

$$\nabla X_{(i)} \in \operatorname{span}\{\nabla D_i\}, \quad i \in [3],$$

where $\operatorname{span}\{\nabla D_i\}$ *represents the linear space spanned by the columns of matrix* ∇D_i. *The notation* $\nabla X_{(i)} \in \operatorname{span}\{\nabla D_i\}$ *indicates that all column vectors of* $\nabla X_{(i)}$ *lie within* $\operatorname{span}\{\nabla D_i\}$.

Theorem 3 implies that the piecewise smooth structure of a tensor can be represented by the factor subspace smoothness along each mode. By integrating this concept with the manifold assumption, which states that data points close to each other in a local neighborhood share similar properties [44], we develop the following generalized hyper-Laplacian regularization:

$$\frac{1}{2}\sum_{m_1}\sum_{m_2}\|D_i(m_1,:)-D_i(m_2,:)\|_p^p\, W^i_{m_1m_2}=\|G_iD_i\|_p^p\,, \tag{10}$$

where W^i be the weight matrix given by

$$W^i_{m_1m_2}=\begin{cases}\exp\left(-\frac{\|X_{(i)}(m_1,:)-X_{(i)}(m_2,:)\|_F^2}{\sigma^2}\right), & \text{if } X_{(i)}(m_1,:) \text{ and } X_{(i)}(m_2,:) \text{ are neighbors,}\\ 0, & \text{otherwise.}\end{cases}$$

The generalized hyper-Laplacian matrix $G_i\in\mathbb{R}^{n_i(n_i-1)/2\times n_i}$ is defined as

$$G_i(g,m_1)=-G_i(g,m_2)=\begin{cases}\sqrt[p]{W^i(m_1,m_2)}, & \text{if } g=(m_1-1)n_i+m_2-m_1(m_1+1)/2,\\ 0, & \text{otherwise.}\end{cases}$$

Example 2 If $n_i=4$, then

$$G_i=\begin{pmatrix}\sqrt[p]{W^i_{12}} & -\sqrt[p]{W^i_{12}} & 0 & 0\\ \sqrt[p]{W^i_{13}} & 0 & -\sqrt[p]{W^i_{13}} & 0\\ \sqrt[p]{W^i_{14}} & 0 & 0 & -\sqrt[p]{W^i_{14}}\\ 0 & \sqrt[p]{W^i_{23}} & -\sqrt[p]{W^i_{23}} & 0\\ 0 & \sqrt[p]{W^i_{24}} & 0 & -\sqrt[p]{W^i_{24}}\\ 0 & 0 & \sqrt[p]{W^i_{34}} & -\sqrt[p]{W^i_{34}}\end{pmatrix}.$$

Remark 5 By applying (10), we can simultaneously mine the local structures of the three directions hidden within $\mathcal{X}$. Specifically, $\|G_1D_1\|_p^p$ and $\|G_2D_2\|_p^p$ jointly capture the spatial appearance, while $\|G_3D_3\|_p^p$ encodes the inherent temporal consistency.

Remark 6 Setting $p=2$ and $G_i=K^i-W^i$ [33] with

$$K^i(m_1,m_2)=\begin{cases}\sum_{m_2}W^i_{m_1m_2}, & \text{if } m_1=m_2,\\ 0, & \text{otherwise,}\end{cases}$$

in (10) leads to

$$\|G_iD_i\|_p^p=\frac{1}{2}\operatorname{tr}\left(D_i^TG_iD_i\right)$$

as studied in [11, 31, 43], which is a hyper-Laplacian regularization term.

Remark 7 Setting $p=1$ and

$$G^i=\begin{pmatrix}1 & -1 & 0 & \cdots & 0 & 0\\ 0 & 1 & -1 & \cdots & 0 & 0\\ 0 & 0 & 1 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & 1 & -1\end{pmatrix}$$

in (10) as studied in [25], which is a total variation (TV) regularization term.

Remark 8 Setting $p = 2$ and

$$G^i = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 & 0 \\ -0.5 & 1 & -0.5 & \cdots & 0 & 0 \\ 0 & -0.5 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -0.5 \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix}$$

in (10) as studied in [36].

Based on the above analysis, and considering that the entries in a spectral vector are corrupted by Gaussian noise, the proposed data-driven tensor dictionary learning (DTDL) model for aligning and removing noise a batch of linearly correlated images is formulated as

$$\min_{\mathcal{L}, D_i, \mathcal{E}, \Gamma} \|\mathcal{L}\|_{\circledast}^{\psi} + \lambda_1 \|\mathcal{E}\|_1^{\psi} + \lambda_2 \sum_{i=1}^{3} \|G_i D_i\|_p^p + \frac{\beta}{2} \|\mathcal{L} \times_1 D_1 \times_2 D_2 \times_3 D_3 + \mathcal{E} - \mathcal{Y} \circ \Gamma\|_F^2 . \tag{11}$$

Remark 9 Since the sum of the nuclear norm (SNN) [13] can destroy the data structure and incur high computation [35, 45], we adopt the TNN norm to approximate the low rankness of the tensor $\mathcal{L}$. At the same time, considering that the ℓ_1-norm may introduce bias in the estimators [5], we employ a class of nonconvex functions to achieve a better approximation of the ℓ_0-norm.

Remark 10 In our proposed model DTDL, two core components are emphasized. First, the underlying tensor is decomposed into a coefficient tensor of small size and three compact dictionary matrices, enabling a more efficient representation. Second, the model incorporates a generalized hyper-Laplacian regularization to preserve the intrinsic local structures embedded in the underlying tensor, as captured by the dictionary framework.

4 Optimization Procedure for DTDL

In this section, to solve the proposed DTDL model, we develop a generalized Gauss-Newton algorithm [1]. Then, a proximal linearized alternating direction method of multipliers (ADMM) algorithm is designed to handle the subproblem that arises from the Gauss-Newton method.

4.1 Gauss-Newton Algorithm for DTDL

One of the challenges in solving the optimal alignment problem (11) is the highly nonlinear equality constraint that involves the domain transformations Γ. To handle this nonlinearity, a common technique is to linearize the constraint around the current estimate of the transformation parameters, especially when the changes in Γ are small or incremental [16]. Specifically, we derive the first-order Taylor approximation of $\mathcal{Y} \circ \Gamma$ at Γ^0 as follow:

$$\mathcal{Y} \circ \Gamma \approx \mathcal{Y} \circ \left(\Gamma^0 + \Delta\Gamma\right) \approx \mathcal{Y} \circ \Gamma^0 + \text{fold}_3 \left(\left(\sum_{k=1}^{n_3} J_k \Delta\Gamma \epsilon_k \epsilon_k^T \right)^T \right),$$

where $\Gamma^0 = [\tau_1^0, \tau_2^0, \ldots, \tau_{n_3}^0]$ and J_k represents the Jacobian of $Y^{(k)}$ with respect to the transformation parameters τ_k^0. Then, problem (11) can be relaxed to the following optimization problem:

$$\begin{aligned}\min_{\mathcal{L}, D_i, \mathcal{E}, \Delta\Gamma} & \|\mathcal{L}\|_{\circledast}^{\psi} + \lambda_1 \|\mathcal{E}\|_1^{\psi} + \lambda_2 \sum_{i=1}^{3} \|G_i D_i\|_p^p \\ & + \frac{\beta}{2} \left\| \mathcal{L} \times_1 D_1 \times_2 D_2 \times_3 D_3 + \mathcal{E} - \mathcal{Y} \circ \Gamma^0 - \Delta\tilde{\Gamma} \right\|_F^2, \end{aligned} \tag{12}$$

where $\Delta\tilde{\Gamma} \doteq \text{fold}_3((\sum_{k=1}^{n_3} J_k \Delta\Gamma \epsilon_k \epsilon_k^T)^T)$. Based on the preceding analysis, Algorithm 1 presents the iterative process of the generalized Gauss-Newton algorithm. For a more comprehensive understanding of the algorithm and its intricacies, further insights and explanations are available in [16, 21].

Algorithm 1 A generalized Gauss-Newton method to solve DTDL

Require: Tensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and parameters λ_1, λ_2.
Initialize: $\Gamma^0 = [\tau_1^0, \tau_2^0, \ldots, \tau_{n_3}^0]$.
while not converge **do**
***Step* 1**. Update Jacobian matrices J_k according to

$$J_k = \frac{\partial}{\partial x} \left(\frac{\text{vec}(\mathcal{Y}(:,:,k) \circ x)}{\| \text{vec}(\mathcal{Y}(:,:,k) \circ x) \|_F} \right) \Bigg|_{x=\tau_k^d}, \ k \in [n_3].$$

***Step* 2**. Wrap and normalize each frontal slice of $\mathcal{Y} \circ \Gamma^d$ according to

$$\left(\mathcal{Y} \circ \Gamma^d\right)(:,:,k) = \frac{\mathcal{Y}(:,:,k) \circ \tau_k^d}{\left\| \mathcal{Y}(:,:,k) \circ \tau_k^d \right\|_F}, \ k \in [n_3].$$

***Step* 3**. Update $(\mathcal{L}, D_i, \mathcal{E}, \Delta\Gamma)$ according to (12).
***Step* 4**. Update the domain transformations Γ^{d+1} according to $\Gamma^{d+1} = \Gamma^d + \Delta\Gamma$.
Let $d := d + 1$ and go to ***Step* 1**.
end while
Ensure: $\mathcal{L}^\star$, $D_i^\star$, $\mathcal{E}^\star$, $\Gamma^\star$.

4.2 Proximal Linearized ADMM Algorithm for (12)

In this part, the proximal linearized alternating direction method of multipliers (ADMM) algorithm is applied to solve the subproblem (12). To facilitate the efficient separation of variables, we introduce matrices C_i, then the augmented Lagrangian function can be written as:

$$\begin{aligned}\mathbb{L}(\mathcal{L}, \mathcal{D}, \mathcal{C}, \mathcal{E}, \Delta\Gamma; \mathcal{Q}) := & \|\mathcal{L}\|_{\circledast}^{\psi} + \lambda_1 \|\mathcal{E}\|_1^{\psi} \\ & + \sum_{i=1}^{3} \left(\lambda_2 \|C_i\|_p^p + \langle Q_i, G_i D_i - C_i \rangle + \frac{\alpha}{2} \|G_i D_i - C_i\|_F^2 \right) \\ & + \frac{\beta}{2} \left\| \mathcal{L} \times_1 D_1 \times_2 D_2 \times_3 D_3 + \mathcal{E} - \mathcal{Y} \circ \Gamma^0 - \Delta\tilde{\Gamma} \right\|_F^2, \end{aligned} \tag{13}$$

where $\mathcal{D} = (D_1, D_2, D_3)$, $\mathcal{C} = (C_1, C_2, C_3)$ and $\mathcal{Q} = (Q_1, Q_2, Q_3)$.

For convenience of notation, let $D_i^t = (D_1^{t+1}, \ldots, D_{i-1}^{t+1}, D_i, D_{i+1}^t, \ldots, D_3^t)$, $D_{-i}^t = (D_1^{t+1}, \ldots, D_{i-1}^{t+1}, D_{i+1}^t, \ldots, D_3^t)$, $\mathcal{L} \times \mathcal{D} = \mathcal{L} \times_1 D_1 \times_2 D_2 \times_3 D_3$ and $\mathcal{L} \times^T \mathcal{D} = \mathcal{L} \times_1 D_1^T \times_2 D_2^T \times_3 D_3^T$. Using the proximal linearized ADMM framework, we update each variable in (13) sequentially while keeping the others fixed.

– Computing $\mathcal{L}^{t+1}$: The subproblem of $\mathcal{L}$ is

$$\min_{\mathcal{L}} \ \|\mathcal{L}\|_{\circledast}^{\psi} + \beta h(\mathcal{L}), \tag{14}$$

where $\mathcal{P}(\mathcal{L}) = \mathcal{L} \times \mathcal{D}^t + \mathcal{E}^t - \mathcal{Y} \circ \Gamma^0 - \Delta\tilde{\Gamma}^t$ and $h(\mathcal{L}) = \frac{1}{2}\|\mathcal{P}(\mathcal{L})\|_F^2$. By linearizing the term $h(\mathcal{L})$ in the objective function of (14) at the current iterate point $\mathcal{L}^t$, $\mathcal{L}^{t+1}$ can be solved by the following minimization problem as:

$$\begin{aligned}
\mathcal{L}^{t+1} &= \arg\min_{\mathcal{L}} \ \|\mathcal{L}\|_{\circledast}^{\psi} + \beta\left\langle \mathcal{P}\left(\mathcal{L}^t\right) \times^T \mathcal{D}^t, \mathcal{L}\right\rangle + \frac{\beta\kappa^t}{2}\left\|\mathcal{L} - \mathcal{L}^t\right\|_F^2 + \frac{\delta}{2}\left\|\mathcal{L} - \mathcal{L}^t\right\|_F^2 \\
&= \arg\min_{\mathcal{L}} \ \|\mathcal{L}\|_{\circledast}^{\psi} + \left\langle \beta\mathcal{P}\left(\mathcal{L}^t\right) \times^T \mathcal{D}^t, \mathcal{L}\right\rangle + \frac{\beta\kappa^t + \delta}{2}\left\|\mathcal{L} - \mathcal{L}^t\right\|_F^2 \\
&= \arg\min_{\mathcal{L}} \ \|\mathcal{L}\|_{\circledast}^{\psi} + \frac{\beta\kappa^t + \delta}{2}\left\|\mathcal{L} - \left(\mathcal{L}^t - \frac{\beta\mathcal{P}\left(\mathcal{L}^t\right) \times^T \mathcal{D}^t}{\beta\kappa^t + \delta}\right)\right\|_F^2 \\
&= \mathrm{Prox}_{\frac{1}{\beta\kappa^t+\delta}\|\cdot\|_{\circledast}^{\psi}}\left(\mathcal{L}^t - \frac{\beta\mathcal{P}\left(\mathcal{L}^t\right) \times^T \mathcal{D}^t}{\beta\kappa^t + \delta}\right),
\end{aligned} \tag{15}$$

where $\delta > 0$ and $\kappa^t := \max\{1e\text{-}3, \prod_{i=1}^3 \|D_i^t\|^2\}$ is a Lipschitz constant of $\nabla h(\mathcal{L})$.

– Computing $\mathcal{D}^{t+1}$: The subproblem of each D_i is

$$\begin{aligned}
\min_{D_i} \ &\left\langle Q_i^t, G_i D_i - C_i^t\right\rangle + \frac{\alpha^t}{2}\left\|G_i D_i - C_i^t\right\|_F^2 + \frac{\beta}{2}\left\|\mathcal{L}^{t+1} \times \mathcal{D}_i + \mathcal{E}^t - \mathcal{Y} \circ \Gamma^0 - \Delta\tilde{\Gamma}^t\right\|_F^2 \\
&+ \frac{\delta}{2}\left\|D_i - D_i^t\right\|_F^2,
\end{aligned}$$

which is equivalent to

$$\min_{D_i} \ \frac{\alpha^t}{2}\left\|G_i D_i - C_i^t + \frac{1}{\alpha^t} Q_i^t\right\|_F^2 + \frac{\beta}{2}\left\|D_i N_{(i)} + M_{(i)}\right\|_F^2 + \frac{\delta}{2}\left\|D_i - D_i^t\right\|_F^2, \tag{16}$$

where $\mathcal{M} = \mathcal{E}^t - \mathcal{Y} \circ \Gamma^0 - \Delta\tilde{\Gamma}^t$ and $\mathcal{N} = \mathcal{L}^{t+1} \times \mathcal{D}_{-i}^t$. Taking the gradient of the (16) with respect to D_i and setting it to zero, we have

$$\left(\alpha^t G_i^T G_i + \frac{\delta}{2} I\right) D_i + D_i\left(\beta N_{(i)} N_{(i)}^T + \frac{\delta}{2} I\right) = -G_i^T\left(Q_i^t - \alpha C_i^t\right) - \beta M_{(i)} N_{(i)}^T - \delta D_i^t.$$

Then, the optimal solution of D_i^{t+1} can be obtained, for example by the Matlab function lyap, i.e.,

$$D_i^{t+1} = \mathrm{lyap}\left(\alpha^t G_i^T G_i + \frac{\delta}{2} I, \beta N_{(i)} N_{(i)}^T + \frac{\delta}{2} I, G_i^T\left(Q_i^t - \alpha C_i^t\right) + \beta M_{(i)} N_{(i)}^T + \delta D_i^t\right). \tag{17}$$

– Computing $\mathcal{C}^{t+1}$: The subproblem of each C_i is

$$\min_{C_i} \ \lambda_2 \|C_i\|_p^p + \frac{\alpha^t}{2}\left\|G_i D_i^{t+1} - C_i + \frac{1}{\alpha^t} Q_i^t\right\|_F^2 + \frac{\delta}{2}\left\|C_i - C_i^t\right\|_F^2. \tag{18}$$

By using the proximity operator of pth power of p-norm [38], we get that

$$C_i^{t+1} = \text{Prox}_{\frac{\lambda_2}{\alpha^t+\delta}\|\cdot\|_p^p}\left(\frac{\alpha^t G_i D_i^{t+1} + Q_i^t + \delta C_i^t}{\alpha^t + \delta}\right). \tag{19}$$

– Computing $\mathcal{E}^{t+1}$: The subproblem of $\mathcal{E}$ is

$$\begin{aligned}&\min_{\mathcal{E}} \ \lambda_1 \|\mathcal{E}\|_1^{\psi} + \frac{\beta}{2}\left\|\mathcal{L}^{t+1} \times \mathcal{D}^{t+1} + \mathcal{E} - \mathcal{Y} \circ \Gamma^0 - \Delta\tilde{\Gamma}^t\right\|_F^2 + \frac{\delta}{2}\left\|\mathcal{E} - \mathcal{E}^t\right\|_F^2 \\ \Leftrightarrow &\min_{\mathcal{E}} \ \lambda_1 \|\mathcal{E}\|_1^{\psi} + \frac{\beta+\delta}{2}\left\|\mathcal{E} - \frac{\beta\left(\mathcal{Y} \circ \Gamma^0 + \Delta\tilde{\Gamma}^t - \mathcal{L}^{t+1} \times \mathcal{D}^{t+1}\right) + \delta\mathcal{E}^t}{\beta+\delta}\right\|_F^2.\end{aligned} \tag{20}$$

Then we get that

$$\mathcal{E}^{t+1} = \text{Prox}_{\frac{\lambda_1}{\beta+\delta}\|\cdot\|_1^{\psi}}\left(\frac{\beta\left(\mathcal{Y} \circ \Gamma^0 + \Delta\tilde{\Gamma}^t - \mathcal{L}^{t+1} \times \mathcal{D}^{t+1}\right) + \delta\mathcal{E}^t}{\beta+\delta}\right). \tag{21}$$

– Computing $\Delta\Gamma^{t+1}$: The subproblem of $\Delta\Gamma$ is

$$\min_{\Delta\Gamma} \ \frac{\beta}{2}\left\|\mathcal{L}^{t+1} \times \mathcal{D}^{t+1} + \mathcal{E}^{t+1} - \mathcal{Y} \circ \Gamma^0 - \Delta\tilde{\Gamma}\right\|_F^2 + \frac{\delta}{2}\left\|\Delta\Gamma - \Delta\Gamma^t\right\|_F^2, \tag{22}$$

where $\Delta\tilde{\Gamma} \doteq \text{fold}_3((\sum_{k=1}^{n_3} J_k \Delta\Gamma \epsilon_k \epsilon_k^T)^T)$. It follows from [21, Theorem 3] that

$$\Delta\Gamma^{t+1} = \sum_{k=1}^{n_3}\left(J_k^T J_k + \delta I\right)^{-1}\left(\delta\Delta\Gamma^t + J_k^T B_{(3)}^T\right)\epsilon_k \epsilon_k^T, \tag{23}$$

where $\mathcal{B} = \mathcal{L}^{t+1} \times \mathcal{D}^{t+1} + \mathcal{E}^{t+1} - \mathcal{Y} \circ \Gamma^0$.

Now, we summary the solving algorithm for (12) in Algorithm 2.

Algorithm 2 Proximal linearized ADMM method to solve (12)

Require: Parameters λ_1, λ_2, β, ρ, δ and function ψ.
Initialize: $\mathcal{L}^0$, $\mathcal{D}^0$, $\mathcal{C}^0$, $\mathcal{E}^0$, $\Delta\Gamma^0$, $\mathcal{Q}^0$, $\mathcal{W}^0$, α^0.
while not converge **do**
***Step* 1**. Update $\mathcal{L}^{t+1}$ according to (15).
***Step* 2**. Update $\mathcal{D}^{t+1}$ according to (17).
***Step* 3**. Update $\mathcal{C}^{t+1}$ according to (19).
***Step* 4**. Update $\mathcal{E}^{t+1}$ according to (21).
***Step* 5**. Update $\Delta\Gamma^{t+1}$ according to (23).
***Step* 5**. Update $\mathcal{Q}^{t+1}$ and α^{t+1} according to

$$Q_i^{t+1} = Q_i^t + \alpha^t\left(G_i D_i^{t+1} - C_i^{t+1}\right), \quad \alpha^{t+1} = \rho\alpha^t, \quad i \in [3]. \tag{24}$$

Let $t := t + 1$ and go to ***Step* 1**.
end while
Ensure: $\mathcal{L}^{t+1}$, $\mathcal{D}^{t+1}$, $\mathcal{E}^{t+1}$, $\Delta\Gamma^{t+1}$.

A summary of the proximal mappings utilized in this paper is provided in "Appendix C", where the specific functional forms are detailed for reference.

4.3 Computational Complexity Analysis

Here we analyze the detailed computational complexity of Algorithm 2. At each iteration, updating $\mathcal{L}$ involves tensor-matrix product costing $\mathcal{O}(n_1n_2n_3r_{123})$ and $\text{Prox}_{\|\cdot\|_{\circledast}^{\psi}}(\cdot)$ costing $\mathcal{O}(r_1r_2r_3\,(\log r_3 + r_{12}))$, where $r_{123} = r_1 + r_2 + r_3$ and $r_{12} = \min\{r_1, r_2\}$. For updating $\mathcal{D}$, we need spend $\mathcal{O}(\sum_{i=1}^{3} n_i^2 \log n_i)$ operations to construct matrices $\{G_i\}_{i=1}^{3}$, $\mathcal{O}(\sum_{i=1}^{3} n_i^3)$ operations for solving Sylvester equation, and $\mathcal{O}(n_1n_2n_3r_{123})$ operations to calculate matrix-matrix product. As for the remaining steps, their computation costs can be ignored since they contain only the basic operations. Thus, the computation complexity of Algorithm 2 is $\mathcal{O}(n_1n_2n_3r_{123} + \sum_{i=1}^{3} n_i^3) \approx \mathcal{O}(n_1n_2n_3r_{123})$. By comparison, the costs of ℓ_p+ADMM [41], which employ nuclear norm minimization and matrix factorization strategy, is $\mathcal{O}(n_1n_2n_3 \sum_{i=1}^{3} n_i)$ at each iteration. NCALTS [21] considers the transformed TNN and has the cost $\mathcal{O}(n_1n_2n_3(\min\{n_1, n_2\} + n_3))$. Based on transformed tensor-tensor product, TFM-TTP [30] costs at each iteration is $\mathcal{O}(n_1n_2n_3(\hat{r} + n_3))$, where $\hat{r}$ is the estimated tensor tubal rank of $\mathcal{X}$. We can observe that our method is more efficient than ℓ_p+ADMM and NCALTS. It is also slightly more efficient than TFM-TTP, which reduces complexity by replacing $\min\{n_1, n_2\}$ with $\hat{r}$ in the first two modes, while the third mode n_3 remains unchanged and thus contributes a larger factor to the cost.

4.4 Convergence Analysis

This subsection is dedicated to the convergence analysis for Algorithm 2. As follows, we first define some notations for the remainder of this paper.

- $\mathcal{T} := (\mathcal{T}_1, \mathcal{T}_2)$ with $\mathcal{T}_1 = (\mathcal{L}, \mathcal{D}, \mathcal{C}, \mathcal{E}, \Delta\Gamma)$ and $\mathcal{T}_2 = \mathcal{Q}$;
- $\Delta\mathcal{L}^{t+1} = \mathcal{L}^{t+1} - \mathcal{L}^t$, $\Delta\mathcal{D}^{t+1} = \mathcal{D}^{t+1} - \mathcal{D}^t$, $\Delta\mathcal{C}^{t+1} = \mathcal{C}^{t+1} - \mathcal{C}^t$, $\Delta\mathcal{E}^{t+1} = \mathcal{E}^{t+1} - \mathcal{E}^t$, $\Delta^2\Gamma^{t+1} = \Delta\Gamma^{t+1} - \Delta\Gamma^t$, $\Delta\mathcal{Q}^{t+1} = \mathcal{Q}^{t+1} - \mathcal{Q}^t$.

We first present a couple of technical lemmas as preparation.

Lemma 1 *For the sequence $\{\mathcal{T}^t\}_{t\in\mathbb{N}^+}$ generated by Algorithm 2, we have*

$$\mathbb{L}\left(\mathcal{T}^{t+1}\right) - \mathbb{L}\left(\mathcal{T}^t\right) \leq -\frac{\delta}{2}\left\|\Delta\mathcal{T}_1^{t+1}\right\|_F^2 + \frac{\rho+1}{2\alpha^t}\left\|\Delta\mathcal{Q}^{t+1}\right\|_F^2. \tag{25}$$

Proof According to (15), one has

$$\left\|\mathcal{L}^{t+1}\right\|_{\circledast}^{\psi} + \beta\left\langle\nabla h\left(\mathcal{L}^t\right), \Delta\mathcal{L}^{t+1}\right\rangle + \frac{\beta\kappa^t}{2}\left\|\Delta\mathcal{L}^{t+1}\right\|_F^2 + \frac{\delta}{2}\left\|\Delta\mathcal{L}^{t+1}\right\|_F^2 \leq \left\|\mathcal{L}^t\right\|_{\circledast}^{\psi},$$

combining which with

$$h\left(\mathcal{L}^{t+1}\right) - h\left(\mathcal{L}^t\right) \leq \left\langle\nabla h\left(\mathcal{L}^t\right), \Delta\mathcal{L}^{t+1}\right\rangle + \frac{\kappa^t}{2}\left\|\Delta\mathcal{L}^{t+1}\right\|_F^2,$$

we obtain

$$\mathbb{L}\left(\mathcal{L}^{t+1}, \mathcal{D}^t, \mathcal{C}^t, \mathcal{E}^t, \Delta\Gamma^t; \mathcal{T}_2^t\right) \leq \mathbb{L}\left(\mathcal{L}^t, \mathcal{D}^t, \mathcal{C}^t, \mathcal{E}^t, \Delta\Gamma^t; \mathcal{T}_2^t\right) - \frac{\delta}{2}\left\|\Delta\mathcal{L}^{t+1}\right\|_F^2. \tag{26}$$

Given that $\mathcal{D}^{t+1}$, $\mathcal{C}^{t+1}$, $\mathcal{E}^{t+1}$, and $\Delta\Gamma^{t+1}$ are the minimizers of (16), (18), (20), and (22), respectively, we can deduce that

$$\begin{aligned}
\mathbb{L}\left(\mathcal{L}^{t+1},\mathcal{D}^{t+1},\mathcal{C}^{t},\mathcal{E}^{t},\Delta\Gamma^{t};\mathcal{T}_2^t\right) &\le \mathbb{L}\left(\mathcal{L}^{t+1},\mathcal{D}^{t},\mathcal{C}^{t},\mathcal{E}^{t},\Delta\Gamma^{t};\mathcal{T}_2^t\right)-\frac{\delta}{2}\left\|\Delta\mathcal{D}^{t+1}\right\|_F^2,\\
\mathbb{L}\left(\mathcal{L}^{t+1},\mathcal{D}^{t+1},\mathcal{C}^{t+1},\mathcal{E}^{t},\Delta\Gamma^{t};\mathcal{T}_2^t\right) &\le \mathbb{L}\left(\mathcal{L}^{t+1},\mathcal{D}^{t+1},\mathcal{C}^{t},\mathcal{E}^{t},\Delta\Gamma^{t};\mathcal{T}_2^t\right)-\frac{\delta}{2}\left\|\Delta\mathcal{C}^{t+1}\right\|_F^2,\\
\mathbb{L}\left(\mathcal{L}^{t+1},\mathcal{D}^{t+1},\mathcal{C}^{t+1},\mathcal{E}^{t+1},\Delta\Gamma^{t};\mathcal{T}_2^t\right) &\le \mathbb{L}\left(\mathcal{L}^{t+1},\mathcal{D}^{t+1},\mathcal{C}^{t+1},\mathcal{E}^{t},\Delta\Gamma^{t};\mathcal{T}_2^t\right)-\frac{\delta}{2}\left\|\Delta\mathcal{E}^{t+1}\right\|_F^2,\\
\mathbb{L}\left(\mathcal{L}^{t+1},\mathcal{D}^{t+1},\mathcal{C}^{t+1},\mathcal{E}^{t+1},\Delta\Gamma^{t+1};\mathcal{T}_2^t\right) &\le \mathbb{L}\left(\mathcal{L}^{t+1},\mathcal{D}^{t+1},\mathcal{C}^{t+1},\mathcal{E}^{t+1},\Delta\Gamma^{t};\mathcal{T}_2^t\right)-\frac{\delta}{2}\left\|\Delta^2\Gamma^{t+1}\right\|_F^2.
\end{aligned}$$

Combining this with (26) gives

$$\mathbb{L}\left(\mathcal{T}_1^{t+1};\mathcal{T}_2^t\right)\le\mathbb{L}\left(\mathcal{T}_1^{t};\mathcal{T}_2^t\right)-\frac{\delta}{2}\left\|\Delta\mathcal{T}_1^{t+1}\right\|_F^2. \tag{27}$$

By the $\mathcal{Q}$ update in (24), we have

$$\begin{aligned}
\mathbb{L}\left(\mathcal{T}_1^{t+1};\mathcal{T}_2^{t+1}\right)&=\mathbb{L}\left(\mathcal{T}_1^{t+1};\mathcal{T}_2^{t}\right)+\frac{\alpha^{t+1}+\alpha^t}{2\left(\alpha^t\right)^2}\left\|\Delta\mathcal{Q}^{t+1}\right\|_F^2\\
&=\mathbb{L}\left(\mathcal{T}_1^{t+1};\mathcal{T}_2^{t}\right)+\frac{\rho+1}{2\alpha^t}\left\|\Delta\mathcal{Q}^{t+1}\right\|_F^2.
\end{aligned}$$

Combining this with (27), we have

$$\mathbb{L}\left(\mathcal{T}^{t+1}\right)-\mathbb{L}\left(\mathcal{T}^{t}\right)\le-\frac{\delta}{2}\left\|\Delta\mathcal{T}_1^{t+1}\right\|_F^2+\frac{\rho+1}{2\alpha^t}\left\|\Delta\mathcal{Q}^{t+1}\right\|_F^2, \tag{28}$$

which completes the proof. □

Lemma 2 *Assume that the sequence $\{\mathcal{T}_1^t\}_{t\in\mathbb{N}^+}$ generated by Algorithm 2 is bounded. Then the following statements hold:*

(i) The sequence $\{\mathcal{T}^t\}_{t\in\mathbb{N}^+}$ is bounded.
(ii) $\lim_{t\to+\infty}\Delta\mathcal{T}_1^{t+1}=0$.

Proof (*i*) We only need to prove that the sequence $\{\mathcal{T}_2^t\}_{t\in\mathbb{N}^+}$, specifically $\{Q_i^t\}_{t\in\mathbb{N}^+}$ for all $i\in[3]$, is bounded. By first-order necessary optimality condition of (18), one has

$$\begin{aligned}
0&\in\lambda_2\partial\left\|C_i^{t+1}\right\|_p^p+\alpha^t\left(C_i^{t+1}-G_iD_i^{t+1}-\frac{1}{\alpha^t}Q_i^t\right)+\delta\left(C_i^{t+1}-C_i^t\right)\\
&=\lambda_2\partial\left\|C_i^{t+1}\right\|_p^p-Q_i^{t+1}-\delta\left(C_i^{t+1}-C_i^t\right),
\end{aligned} \tag{29}$$

where the last equality follows from (24). Given the above equation and the boundedness of $\{C_i^t\}_{t\in\mathbb{N}^+}$, it remains to show that $\partial\|C_i^{t+1}\|_p^p$ is also bounded. We analyze this in three cases:

Case 1. $p\in(0,1)$. In order to overcome the singularity of $(|x|^p)'=p\,\mathrm{sign}(x)/|x|^{1-p}$ near ∞ as x near 0, we consider for $0<\epsilon\ll1$ the approximation:

$$\left(|x|^p\right)'\approx\frac{p\,\mathrm{sign}(x)}{\max\left\{\epsilon^{1-p},|x|^{1-p}\right\}}.$$

Thus, $\|\partial\|C_i^{t+1}\|_p^p\|_F\le n_i(n_i-1)r_ip/(2\epsilon^{1-p})$ is bounded.

Case 2. $p=1$. $\|\partial\|C_i^{t+1}\|_p^p\|_F\le n_i(n_i-1)r_i/2$ is bounded since $|\partial|x||\le1$.

Case 3. $p\in(1,2]$. $\|\partial\|C_i^{t+1}\|_p^p\|_F=\sum_{m_1,m_2}p|C_i^{t+1}(m_1,m_2)|^{p-1}$ is bounded since C_i^{t+1} is bounded.

Combining all cases, we conclude that $\partial\|C_i^{t+1}\|_p^p$ is indeed bounded. This completes the proof of result (i) in this lemma.

(ii) By summing up (25) from $t=1$ to some $T\geq 1$, we notice that

$$\frac{\delta}{2}\sum_{t=1}^{T}\left\|\Delta\mathcal{T}_1^{t+1}\right\|_F^2\leq\mathbb{L}\left(\mathcal{T}^1\right)-\mathbb{L}\left(\mathcal{T}^{T+1}\right)+\sum_{t=1}^{T}\frac{\rho+1}{2\alpha^t}\left\|\Delta\mathcal{Q}^{t+1}\right\|_F^2. \tag{30}$$

Given that $\{\mathcal{Q}^t\}_{t\in\mathbb{N}^+}$ is bounded, there exists a constant $M>0$ such that $\max\{\|\Delta\mathcal{Q}^t\|_F, \|\mathcal{Q}^t\|_F\}\leq M$ holds for any t. Observing that $\alpha^t=\rho\alpha^{t-1}=\ldots=\rho^{t-1}\alpha^1$, we have

$$\begin{aligned}\sum_{t=1}^{T}\frac{\rho+1}{2\alpha^t}\left\|\Delta\mathcal{Q}^{t+1}\right\|_F^2&\leq M^2\sum_{t=1}^{T}\frac{\rho+1}{2\alpha^t}\\&\leq M^2\sum_{t=1}^{+\infty}\frac{\rho+1}{2\alpha^t}=M^2\sum_{t=1}^{+\infty}\frac{\rho+1}{2\rho^{t-1}\alpha^1}=\frac{\rho(\rho+1)M^2}{2\alpha^1(\rho-1)}.\end{aligned} \tag{31}$$

Recalling the expression for $\mathbb{L}\left(\mathcal{T}^t\right)$, we get

$$\begin{aligned}\mathbb{L}\left(\mathcal{T}^t\right)&=\left\|\mathcal{L}^t\right\|_{\circledast}^{\psi}+\lambda_1\left\|\mathcal{E}^t\right\|_1^{\psi}\\&\quad+\sum_{i=1}^{3}\left(\lambda_2\left\|C_i^t\right\|_p^p+\frac{\alpha^t}{2}\left\|G_iD_i^t-C_i^t+\frac{Q_i^t}{\alpha^t}\right\|_F^2-\frac{\left\|Q_i^t\right\|_F^2}{2\alpha^t}\right)+\beta h\left(\mathcal{L}^t\right)\\&\geq-\sum_{i=1}^{3}\frac{\left\|Q_i^t\right\|_F^2}{2\alpha^t}\geq-\frac{3M^2}{2\alpha^1}.\end{aligned} \tag{32}$$

By combining inequalities (30), (31), and (32), we obtain

$$\frac{\delta}{2}\sum_{t=1}^{T}\left\|\Delta\mathcal{T}_1^{t+1}\right\|_F^2\leq\mathbb{L}\left(\mathcal{T}^1\right)+\frac{3M^2}{2\alpha^1}+\frac{\rho(\rho+1)M^2}{2\alpha^1(\rho-1)}.$$

Taking the limit as $T\to+\infty$ in the above inequality, we conclude that $\sum_{t=1}^{+\infty}\|\Delta\mathcal{T}_1^{t+1}\|_F^2\leq+\infty$. This implies that $\lim_{t\to+\infty}\Delta\mathcal{T}_1^{t+1}=0$, thus completing the proof of this statement. □

In terms of the convergence of Algorithm 2 for problem (11), we have the following main results.

Theorem 4 *Let $\{\mathcal{T}^t\}_{t\in\mathbb{N}^+}$ be a sequence generated by Algorithm 2. Suppose that the sequence $\{\mathcal{T}_1^t\}_{t\in\mathbb{N}^+}$ is bound. Then any accumulation point of the sequence $\{\mathcal{T}^t\}_{t\in\mathbb{N}^+}$ is a Karush–Kuhn–Tucker (KKT) point of the optimization problem (12).*

Proof The proof proceeds as follows:

Step 1: Convergence of the sequence $\{\mathcal{T}^t\}_{t\in\mathbb{N}^+}$. Since $\{\mathcal{T}^t\}_{t\in\mathbb{N}^+}$ is bounded, which is proved in Lemma 2-(i), there exists a subsequence of $\{\mathcal{T}^t\}_{t\in\mathbb{N}^+}$ (also denoted by $\{\mathcal{T}^t\}_{t\in\mathbb{N}^+}$ for simplicity) and $\mathcal{T}^\star$ such that $\lim_{t\to+\infty}\mathcal{T}^t=\mathcal{T}^\star$.

Step 2: Feasibility of $\mathcal{T}^\star$. According to the dual update scheme in (24), one has

$$\lim_{t\to+\infty}\left\|G_iD_i^{t+1}-C_i^{t+1}\right\|_F=\lim_{t\to+\infty}\frac{1}{\alpha^t}\left\|Q_i^{t+1}-Q_i^t\right\|_F=0. \tag{33}$$

This follows that $\lim_{t\to+\infty}(G_iD_i^{t+1}-C_i^{t+1})=0$, and thus $G_iD_i^\star=C_i^\star$ for $i\in[3]$. This indicates that this accumulation point can satisfy the feasible conditions of (12).

Step 3: First-order optimality conditions. For the $\mathcal{L}$-subproblem, the first-order optimality condition states that

$$0 \in \partial \left\| \mathcal{L}^{t+1} \right\|_{\circledast}^{\psi} + \beta \nabla h\left(\mathcal{L}^{t}\right) + \left(\beta \kappa^{t} + \delta\right)\left(\mathcal{L}^{t+1} - \mathcal{L}^{t}\right), \tag{34}$$

combining which with $\lim_{t\to+\infty} \mathcal{L}^{t+1} - \mathcal{L}^{t} = 0$, we obtain

$$0 \in \partial \left\| \mathcal{L}^{\star} \right\|_{\circledast}^{\psi} + \beta \nabla h\left(\mathcal{L}^{\star}\right). \tag{35}$$

For the $\mathcal{D}$-subproblem, the first-order optimality condition states that

$$0 \in \alpha^{t} G_i^{T}\left(G_i D_i^{t+1} - C_i^{t} + \frac{1}{\alpha^{t}} Q_i^{t}\right) + \beta\left(D_i^{t+1} N_{(i)} + M_{(i)}\right) N_{(i)}^{T} + \delta\left(D_i^{t+1} - D_i^{t}\right), \tag{36}$$

where $\mathcal{M} = \mathcal{E}^{t} - \mathcal{Y} \circ \Gamma^{0} - \Delta\tilde{\Gamma}^{t}$ and $\mathcal{N} = \mathcal{L}^{t+1} \times \mathcal{D}_{-i}^{t}$. Due to (24) and Lemma 2-(ii), one has

$$\begin{aligned} &\lim_{t\to+\infty} \alpha^{t} G_i^{T}\left(G_i D_i - C_i^{t} + \frac{1}{\alpha^{t}} Q_i^{t}\right) = \lim_{t\to+\infty} G_i^{T} Q_i^{t} + \alpha^{t}\left(G_i D_i^{t+1} - C_i^{t}\right) \\ &= \lim_{t\to+\infty} G_i^{T} Q_i^{t} + \alpha^{t}\left(G_i D_i^{t+1} - C_i^{t+1}\right) = \lim_{t\to+\infty} G_i^{T} Q_i^{t} + \left(Q_i^{t+1} - Q_i^{t}\right) = G_i^{T} Q_i^{\star}. \end{aligned} \tag{37}$$

Recalling Lemma 2-(ii) and invoking (36), (37), we see that

$$0 \in G_i^{T} Q_i^{\star} + \beta\left(D_i^{\star}\left(\mathcal{L}^{\star} \times \mathcal{D}_{-i}^{\star}\right)_{(i)} + \left(\mathcal{E}^{\star} - \mathcal{Y} \circ \Gamma^{0} - \Delta\tilde{\Gamma}^{\star}\right)_{(i)}\right)\left(\mathcal{L}^{\star} \times \mathcal{D}_{-i}^{\star}\right)_{(i)}^{T}. \tag{38}$$

For the $\mathcal{C}$-subproblem, the first-order optimality condition states that

$$0 \in \lambda_2 \partial \left\| C_i^{t+1} \right\|_{p}^{p} + \alpha^{t}\left(C_i^{t+1} - G_i D_i^{t+1} - \frac{1}{\alpha^{t}} Q_i^{t}\right) + \delta\left(C_i^{t+1} - C_i^{t}\right), \tag{39}$$

combining which with (24) and Lemma 2-(ii), we can easily observe that

$$0 \in \lambda_2 \partial \left\| C_i^{\star} \right\|_{p}^{p} - Q_i^{\star}. \tag{40}$$

For the $\mathcal{E}$-subproblem, the first-order optimality condition states that

$$0 \in \lambda_1 \partial \left\| \mathcal{E}^{t+1} \right\|_{1}^{\psi} + \beta\left(\mathcal{L}^{t+1} \times \mathcal{D}^{t+1} + \mathcal{E}^{t+1} - \mathcal{Y} \circ \Gamma^{0} - \Delta\tilde{\Gamma}^{t}\right) + \delta\left(\mathcal{E}^{t+1} - \mathcal{E}^{t}\right). \tag{41}$$

Letting $t \to +\infty$ in (41) and by Lemma 2-(ii), we obtain

$$0 \in \lambda_1 \partial \left\| \mathcal{E}^{\star} \right\|_{1}^{\psi} + \beta\left(\mathcal{L}^{\star} \times \mathcal{D}^{\star} + \mathcal{E}^{\star} - \mathcal{Y} \circ \Gamma^{0} - \Delta\tilde{\Gamma}^{\star}\right). \tag{42}$$

For the $\Delta\Gamma$-subproblem, the first-order optimality condition states that

$$\begin{aligned} 0 \in \frac{\beta}{2} \partial \left\| \mathcal{L}^{t+1} \times \mathcal{D}^{t+1} + \mathcal{E}^{t+1} - \mathcal{Y} \circ \Gamma^{0} - \mathrm{fold}_3\left(\left(\sum_{k=1}^{n_3} J_k \Delta\Gamma^{t+1} \epsilon_k \epsilon_k^{T}\right)^{T}\right) \right\|_{F}^{2} \\ + \delta\left(\Delta\Gamma^{t+1} - \Delta\Gamma^{t}\right). \end{aligned} \tag{43}$$

Let $t \to +\infty$, we obtain that

$$0 \in \frac{\beta}{2} \partial \left\| \mathcal{L}^{\star} \times \mathcal{D}^{\star} + \mathcal{E}^{\star} - \mathcal{Y} \circ \Gamma^{0} - \mathrm{fold}_3\left(\left(\sum_{k=1}^{n_3} J_k \Delta\Gamma^{\star} \epsilon_k \epsilon_k^{T}\right)^{T}\right) \right\|_{F}^{2}. \tag{44}$$

Step 4: Conclusion. So, by $G_i D_i^\star = C_i^\star$ for $i \in [3]$, (35), (38), (40), (42) and (44), we have that $\mathcal{T}^\star$ is a KKT point of (12). □

5 Experiments

In this section, experiments are conducted to evaluate the performance of the proposed algorithm on a server with an Intel Core i5-12500 H CPU (2.60 GHz) and 16 G memory. The codes of all algorithms are implemented on MATLAB 2022a without any preprocessing for fairness. As a comparison, we use some general methods in the experiments, such as three low rank matrix based methods (i.e., RASL [16], t-GRASTA [6], NQLSD [2]) and three low rank tensor based methods (i.e., ℓ_p+ADMM [41], NCALTS [21], TFM-TTP [30]).

In all experiments, we randomly initialized $\mathcal{L}$ and $\mathcal{D}$ using the matlab command " randn ", and assigned $G_i D_i$ to C_i for $i \in [3]$, while the rest of the variables were assigned to corresponding zero tensors or zero matrices. We describe the parameter settings of our experiments in Subsection 5.3. To ensure the fairness of our results, we run our algorithm 10 times and take the average. We use Algorithm 1 and Algorithm 2 with different stopping criteria and maximum number of iterations. For Algorithm 1, we stop when $\|\mathcal{L}^{d+1} - \mathcal{L}^d\|_\infty \leq$ tol and $\|D_i^{d+1} - D_i^d\|_\infty \leq$ tol for $i \in [3]$, or when the number of iterations reaches 60. For Algorithm 2, we stop when $\|\mathcal{L}^{t+1} \times \mathcal{D}^{t+1} + \mathcal{E}^{t+1} - \mathcal{Y} \circ \Gamma^0 - \Delta\tilde{\Gamma}^{t+1}\|_\infty \leq$ tol and $\|G_i D_i^{t+1} - C_i^{t+1}\|_\infty \leq$ tol for $i \in [3]$, or when the number of iterations reaches 300. We conduct our experiments using the MCP function as ψ and setting tol $= 5e$-3 for all cases. In the case of noise, we choose $p = 1$, while in the case of no noise, we choose $p = 2$. As for the compared methods, their parameters are manually tuned for best performances.

5.1 Image Alignment

5.1.1 AR Face Database

In this part, to evaluate our methods, we conduct image alignment on AR Face database[1] [15]. The dataset consists of more than 4,000 color photos of 126 individuals' faces (70 males and 56 females). The photos show frontal views of faces with various expressions, lighting conditions, and occlusions (such as sunglasses and scarves). We choose 8 clear and well-aligned images of the subject w-039 and then resize these images to 80×60 as the ground-truth images. We apply Euclidean transformations to each image to generate 20 images with different angles of rotation and translations. The angles of rotation are uniformly distributed in the range $[-2.5, 2.5]$ degrees. The x- and y-translations are uniformly distributed in the range $[-1, 1]$. We then add either 10%-20% salt-and-pepper impulse noise to the 160 transformed images. We crop the central 50×40 region of each face image to form a tensor $\mathcal{Y}$ of size $50 \times 40 \times 160$ for each group of generated images. To achieve an accurate evaluation of the proposed method, the peak signal-to-noise ratio (PSNR) [28], the structural similarity index (SSIM) [28] and the recovery computation time are employed.

Table 1 gives the quantitative metrics and running time (s) of the results by different methods with salt-and-pepper impulse noise at different levels. We can find that the capabilities of RASL, t-GRASTA, and NQLSD are limited and this phenomenon accords with the visual results shown in Fig. 1 and 2. The effectiveness of these three methods is severely affected

[1] http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html

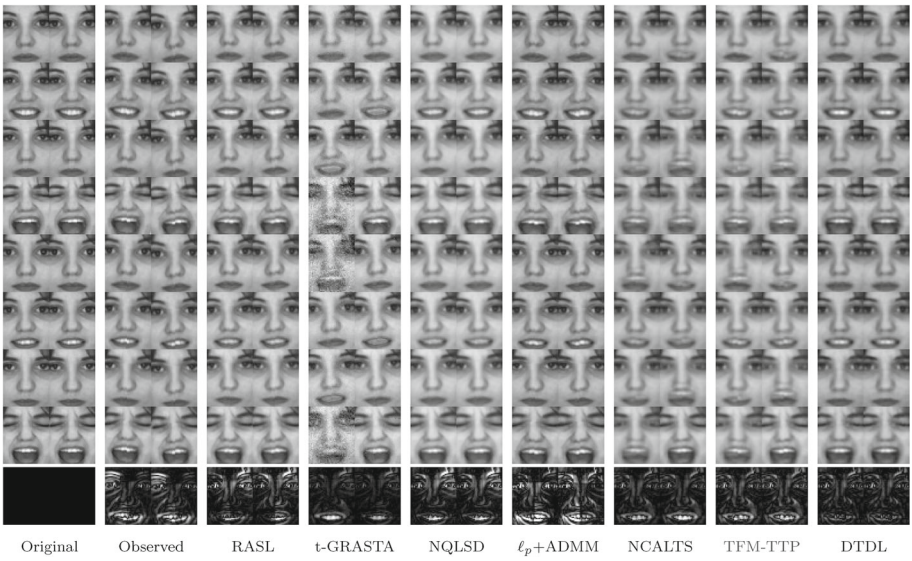

Fig. 1 Image alignment sample of AR Face database. For better visualization, we show the error map (difference from the original) of the second row in the last row

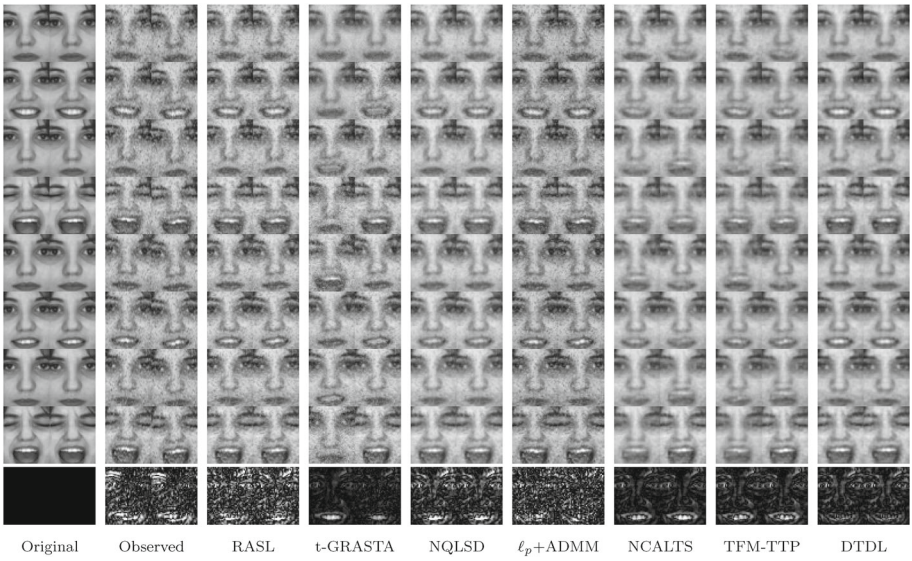

Fig. 2 Image alignment sample of AR Face database under 10% salt-and-pepper impulse noise. For better visualization, we show the error map (difference from the original) of the second row in the last row

Table 1 PSNR, SSIM values and running time in seconds of the result by different methods for the AR Face database with salt-and-pepper impulse noise at different levels

Methods	PSNR			SSIM			Time (s)		
	0	10%	20%	0	10%	20%	0	10%	20%
RASL	23.31	22.62	21.13	0.9921	0.9913	0.9886	15.15	20.55	21.13
t-GRASTA	21.13	20.55	19.96	0.9866	0.9849	0.9839	35.53	35.35	34.28
NQLSD	23.20	23.18	21.85	0.9923	0.9923	0.9901	11.69	13.91	23.69
ℓ_p+ADMM	24.65	23.34	21.34	0.9928	0.9924	0.9887	21.48	26.62	27.19
NCALTS	24.27	23.58	22.18	0.9933	0.9925	0.9905	20.10	23.38	24.56
TFM-TTP	24.79	24.04	22.36	0.9940	0.9933	0.9909	15.51	14.92	14.65
DTDL	26.26	24.77	23.04	0.9961	0.9946	0.9924	7.92	19.04	22.77

by matrixization, which destroys the internal structure of the data. ℓ_p+ADMM, NCALTS and TFM-TTP achieve better metrics, but their performance is still unsatisfactory due to their inability to exploit the spatial and temporal patterns of the data. We also notice that ℓ_p+ADMM is most affected by the level of salt-and-pepper noise; NCALTS and TFM-TTP result in some originally closed mouths in face images being restored as open mouths due to excessive use of low rank approximations. The visual quality of our DTDL outperforms that of other methods, especially for the teeth. They appear more distinct and realistic than those recovered by other methods, as evidenced in the last row. Correspondingly, we can see from Table 1 that our method DTDL achieves the best PSNR and SSIM values. Specifically, in the noise-free case, our method DTDL obtains a PSNR value about 1.47db higher than the second best method, and in the noisy case, our method DTDL obtains a PSNR value about 0.68db higher than the second best method. In terms of computational time, although DTDL exhibits slightly longer runtime in the presence of salt-and-pepper noise, it performs nearly twice as fast as TFM-TTP and significantly outperforms NCALTS and ℓ_p+ADMM in noise-free conditions. Given the substantial improvements in PSNR and SSIM, the minor increase in runtime under salt-and-pepper noise is a reasonable trade-off for the enhanced image quality and structural similarity.

5.1.2 AI Gore Talking Database

In this part, we evaluate our method on the widely used AI Gore talking database [16]. The dataset contains 140 face images that vary in size, alignment, and quality due to changes in pose and illumination. We preprocess the images by cropping them to 80×60 pixels and centering them on the facial regions. The resulting testing tensor for this dataset has a dimension of $80 \times 60 \times 140$.

We present the alignment results and the corresponding running times of different algorithms under two scenarios: noise-free and 10% salt-and-pepper impulse noise, in Figs. 3 and 4, respectively. Figure 3 shows the alignment results for the noise-free case. RASL failed to align the images well, especially in the first row, where the face images are slightly tilted to the right. t-GRASTA aligned the images better, but some facial details were lost, such as the closed mouth in the fourth row that was recovered as open, and the large mouth opening in the third row that was recovered as smaller. This is a drawback of pursuing low rank excessively. NQLSD, ℓ_p+ADMM, NCALTS and TFM-TTP achieved relatively better alignment effects, but they also had some minor flaws. NQLSD and NCALTS lost some details of the mouth

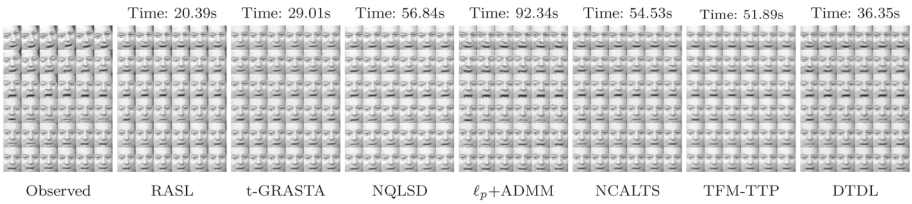

Fig. 3 Image alignment sample of AI Gore talking database

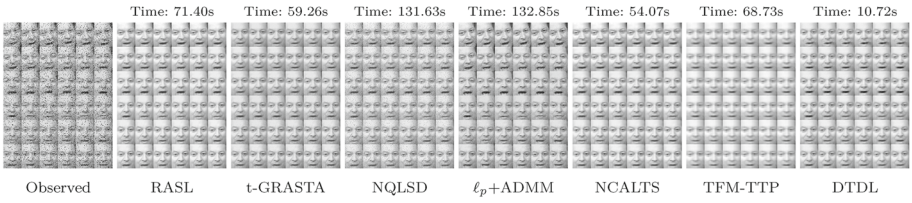

Fig. 4 Image alignment sample of AI Gore talking database under 10% salt-and-pepper impulse noise

area of the aligned face images, while ℓ_p+ADMM and TFM-TTP had a longer running time. Our method DTDL obtained a high-quality alignment result and had a fast running speed. Figure 4 shows the alignment results for the 10% salt-and-pepper impulse noise case. RASL and t-GRASTA had the same defects as in the noise-free case. NQLSD and ℓ_p+ADMM, which performed well in the noise-free image alignment, performed poorly in the noisy image alignment, especially ℓ_p+ADMM, which not only failed to align the images but also introduced a lot of noise. NCALTS had a similar alignment effect in the noisy case as in the noise-free case, except that the aligned images were slightly tilted to the right. Images aligned using TFM-TTP become overly bright, resulting in the loss of facial details. The noise did not affect our method DTDL much, except for some loss of details. In the presence of noise, our method DTDL showed more prominent advantages in terms of alignment quality and running time.

5.1.3 Handwritten Digits Database

In this part, we assess our method using the widely recognized MNIST database of handwritten digits.[2] For our evaluation, we specifically focus on the handwritten digit "3". We selected 100 images, each with a resolution of 28×28 pixels, resulting in a tensor of dimensions $28 \times 28 \times 100$.

Figure 5 illustrates the alignment results and the corresponding running times of various algorithms under noise-free conditions. Unlike regular grayscale images, handwritten digits contain non-zero values only where the digits are present, with all other pixels being zero, which results in strong spatial correlation. However, due to the varying shapes of the digits, the temporal correlation is relatively weak. Imposing low rank constraints alone often leads to significant loss of details, such as font thickness and shape, after alignment. This issue is particularly evident in algorithms like RASL, t-GASTA, NCALTS, and TFM-TTP, where the aligned images tend to look very similar, especially with TFM-TTP. On the other hand, methods such as NQLSD, ℓ_p+ADMM, and our proposed DTDL maintain more details of the

[2] https://yann.lecun.com/exdb/mnist/

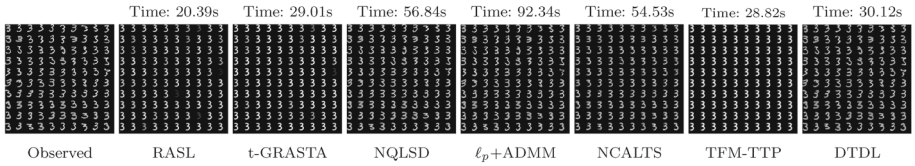

Fig. 5 Image alignment sample of handwritten digits "3" database

digits. Among these three methods, our DTDL method has the shortest runtime. In summary, our proposed method not only improves efficiency but also produces better alignment results.

5.2 Face Recognition

In this subsection, we compare the performance of the algorithms for some face recognition tasks with salt-and-pepper impulse noise. We use two data sets for face recognition, which are described as follows:

- ORL database.[3] This is a widely used data set of face images, containing 400 images of 40 different persons, with 10 images per person. Due to the computational limitation, we resize each image to 40×40 pixels. For each person, we select the first 5 images as training samples, and the rest as testing samples.
- UMIST Face database.[4] This consists of 564 images of 20 people, each covering a range of poses from profile to frontal views. In the experiments, we choose the first 20 images per person, and resize each image to 40×40. For each person, the numbers of training and testing samples are both 10.

We apply Euclidean transformations to each image with a randomly selected angle of rotation from the range $[-2.5, 2.5]$ degrees, and then add 10% salt-and-pepper impulse noise.

To illustrate the process of face recognition, we present a flowchart in Fig. 6, which can be summarized as follows:

Step 1 We apply the six algorithms to recover the low rank structure from the corrupted images.

Step 2 We use the principal component analysis (PCA) on the recovered images to obtain the feature matrix P.

Step 3 We project the original training samples and testing samples onto the feature matrix P to obtain T_p and E_p.

Step 4 We use the sparse representation classification (SRC) [29] algorithm on T_p and E_p to obtain the recognition accuracies of the testing samples.

The recognition accuracies on the two datasets are shown in Fig. 7, from which we can see that our proposed method DTDL achieves the best performance in most cases, and outperforms the other five methods in terms of accuracy and robustness. Specifically, our proposed method DTDL improves the accuracy by about 4% compared to the second best method when the number of principal components is greater than or equal to 40. Moreover, our proposed method DTDL is robust to the number of principal components, as its accuracy remains stable at a high level, while the accuracies of other methods vary with some fluctuations, except for RASL which consistently has a low accuracy. Therefore, our method demonstrates a superior performance for face recognition.

[3] https://www.kaggle.com/datasets/kasikrit/att-database-of-faces

[4] https://www.visioneng.org.uk/datasets/

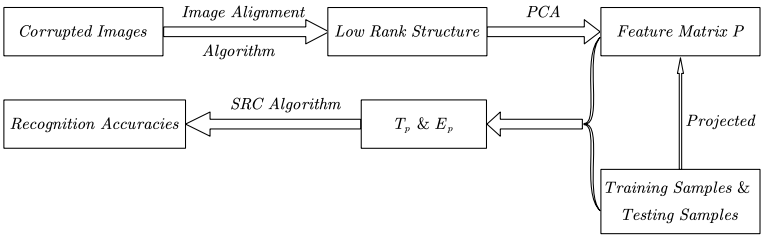

Fig. 6 Flowchart of face recognition

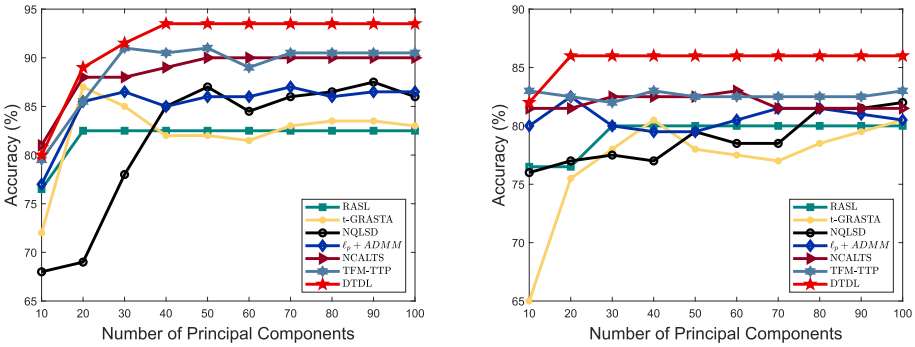

Fig. 7 The contrast results on the two face datasets

Table 2 The result of paired t-test of p-values for comparison of recognition accuracies between DTDL and other methods on the ORL database

Comparison	RASL	t-GRASTA	NQLSD	ℓ_p+ADMM	NCALTS	TFM-TTP
p-value	1.8628×10^{-5}	3.9808×10^{-5}	4.3966×10^{-4}	0.0026	0.0271	0.0148

A statistical test was conducted to quantitatively compare the performance differences among various methods [12]. The grouped cross-validation paired t-test was employed to compare the recognition accuracy of DTDL against six other methods, with a significance level set at 0.05. A p-value of less than 0.05 indicates a significant difference between the two methods. As shown in Table 2, the p-values for comparisons between DTDL and the other methods are all less than 0.05. These results demonstrate that the performance differences between DTDL and the other methods are statistically significant.

5.3 Discussions

In this part, we first discuss the effects of low rank coding and generalized hyper-Laplacian regularization. Next, the influence of different parameters is analyzed. We then explore the convergence behavior of our proposed algorithm with randomly selected initial values. Finally, the limitations and potential failure cases of the proposed approach are discussed. All tests are based on the AR Face database.

Table 3 PSNR, SSIM values and running time in seconds of the result by DTDL under different coefficient tensor codings with salt-and-pepper impulse noise at different levels

Coding versions	PSNR			SSIM			Time (s)		
	0	10%	20%	0	10%	20%	0	10%	20%
$\|\mathcal{L}\|_{\ell_1^\psi}$	24.44	24.41	22.56	0.9924	0.9939	0.9914	31.61	38.99	32.78
$\|\mathcal{L}\|_{\ell_{1,1,2}^\psi}$	25.92	24.41	22.73	0.9956	0.9941	0.9919	10.81	14.93	25.58
$\|\mathcal{L}\|_{\circledast}^\psi$	26.26	24.77	23.04	0.9961	0.9946	0.9924	7.92	19.04	22.77

Table 4 PSNR, SSIM values and running time in seconds of the result by DTDL under different regularization of dictionary with salt-and-pepper impulse noise at different levels

Regularization versions	PSNR			SSIM			Time (s)		
	0	10%	20%	0	10%	20%	0	10%	20%
Non-regularized	24.33	24.21	22.60	0.9937	0.9937	0.9917	27.35	39.71	24.94
Total variation (TV)	21.30	19.96	18.48	0.9860	0.9821	0.9776	21.38	21.82	24.39
Hyper-Laplacian	26.26	24.32	22.43	0.9961	0.9940	0.9913	7.92	16.06	14.81
Generalized hyper-Laplacian	26.26	24.77	23.04	0.9961	0.9946	0.9924	7.92	19.04	22.77

5.3.1 Model Analysis

(1) **Effects of low rank coding used in the DTDL** We evaluate the impact of low rank coding on the proposed DTDL by comparing it under different coding versions, including $\|\mathcal{L}\|_{\ell_1} = \sum_{ijk} |\mathcal{L}_{ijk}|$ and $\|\mathcal{L}\|_{\ell_{1,1,2}} = \sum_{ij} \|\mathcal{L}(i, j, :)\|_F$. For a fair comparison, we replace the above two coefficient tensor norms with their corresponding nonconvex versions: $\|\mathcal{L}\|_{\ell_1^\psi} = \sum_{ijk} \psi(|\mathcal{L}_{ijk}|)$ and $\|\mathcal{L}\|_{\ell_{1,1,2}^\psi} = \sum_{ij} \psi(\|\mathcal{L}(i, j, :)\|_F)$. Table 3 lists the average recovery PSNR, SSIM values and the corresponding average running times by DTDL under different coefficient tensor codings with salt-and-pepper impulse noise at different levels. The proposed low rank coding can be seen to achieve the best performance. Therefore, we suggest exploiting the low rankness of $\mathcal{L}$ to improve the performance.

(2) **Effects of generalized hyper-Laplacian regularization used in the DTDL** We evaluate the impact of generalized hyper-Laplacian regularization on the proposed DTDL by comparing it under different regularization versions, including non-regularized, total variation (TV) regularization, and hyper-Laplacian regularization, as shown in Table 4. Table 4 shows that TV regularization has the worst performance. In the noise-free case, we set $p = 2$ in DTDL, and the generalized hyper-Laplacian regularization reduces to hyper-Laplacian regularization, resulting in the same performance for both methods. In the noisy case, we set $p = 1$ in DTDL, and the generalized hyper-Laplacian regularization outperforms hyper-Laplacian regularization, especially at higher noise levels. In comparison to the non-regularized version, generalized hyper-Laplacian regularization not only demonstrates improved numerical performance but also significantly reduces running time. Therefore, we recommend using the generalized hyper-Laplacian regularization to enhance the performance.

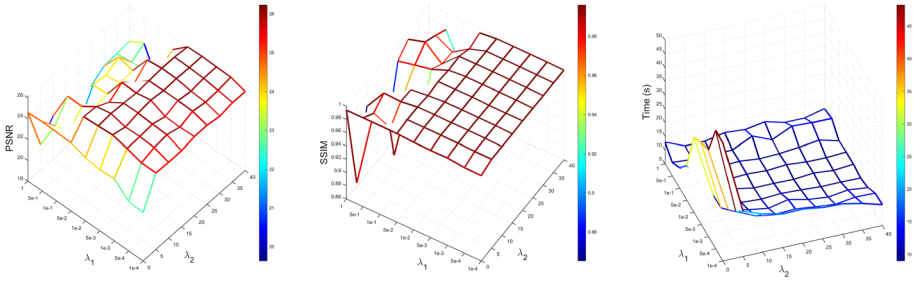

Fig. 8 Surfaces of average PSNR, SSIM values and running time in seconds of the result by our method with different λ_1 and λ_2

5.3.2 Parameter Analysis

There are two regularization parameters λ_1, λ_2 and six algorithm parameters β, α, δ, r_1, r_2, r_3 that need to be manually set to make the algorithm achieve optimal performance. In Algorithm 2, we initialize α as $5e$-3 and increase it gradually as the iteration progresses. To ensure that α approaches infinity, we multiply it by 1.2 at the 15th, 20th, and 25th iterations, and by 5 at every iteration from the 30th iteration onwards, until the convergence condition is met. For both δ and β, we use the same iteration method as for α, except that we fix δ at its maximum value of $1e$-5 when it reaches this limit, and we fix β at its maximum value of $1e10$ when it reaches this limit. The parameter (r_1, r_2, r_3) affects the recovery performance for different image datasets, and we select it to achieve the best recovery results in the experiments.

The following provides a general guideline for determining r_1, r_2, and r_3: The first two dimensions of the original tensor encode the spatial information of the images. When n_1 and n_2 are relatively small, indicating limited spatial information, choosing larger values for r_1 and r_2-potentially exceeding n_1 and n_2-can better preserve this information. Conversely, when n_1 and n_2 are large, smaller values for r_1 and r_2 are sufficient to maintain the spatial structure. The third dimension of the tensor corresponds to the collection of images and is typically large. However, as these images are often similar, r_3 can be set significantly smaller than n_3.

The performance is compared for different values of λ_1 and λ_2. λ_1 is tested with candidates $\{1e\text{ - }4, 5e\text{ - }4, 1e\text{ - }3, 5e\text{ - }3, 1e\text{ - }2, 5e\text{ - }2, 1e\text{ - }1, 5e\text{ - }1, 1\}$ while λ_2 varies from 0 to 40 with a step size 5. As shown in Fig. 8, the algorithms perform well over a wide range of values of λ_1 and λ_2. Specifically, when $\lambda_1 \leq 1e\text{ - }2$, the PSNR and SSIM values for $\lambda_2 > 0$ are not only relatively stable, but also work well, and higher than the PSNR and SSIM values for $\lambda_2 = 0$. This indirectly demonstrates the effectiveness of the generalized hyper-Laplacian regularization we proposed. In terms of time consumption, when $\lambda_1 \in [5e\text{ - }3, 1e\text{ - }1], \lambda_2 > 0$, the algorithm runs faster, and is basically stable at around 10 s. Based on this analysis, we set $\lambda_1 = 5e$-3 and $\lambda_2 = 20$ in this article.

5.3.3 Analysis of Convergence with Random Initializations

We also examined the robustness of our proposed algorithm with randomly selected initial values for $\mathcal{L}$ and $\mathcal{D}$ in terms of convergence behavior. Figure 9 presents the relative change of Γ, $\mathcal{X} = \mathcal{L} \times_1 D_1 \times_2 D_2 \times_3 D_3$, $\mathcal{P} = [\mathcal{E}(:); \Delta\Gamma(:)]$, and PSNR with respect to iteration numbers for different random initializations. Our algorithm includes outer iterations (denoted as d) and inner iterations (denoted as t). According to the results in the figure, there are three

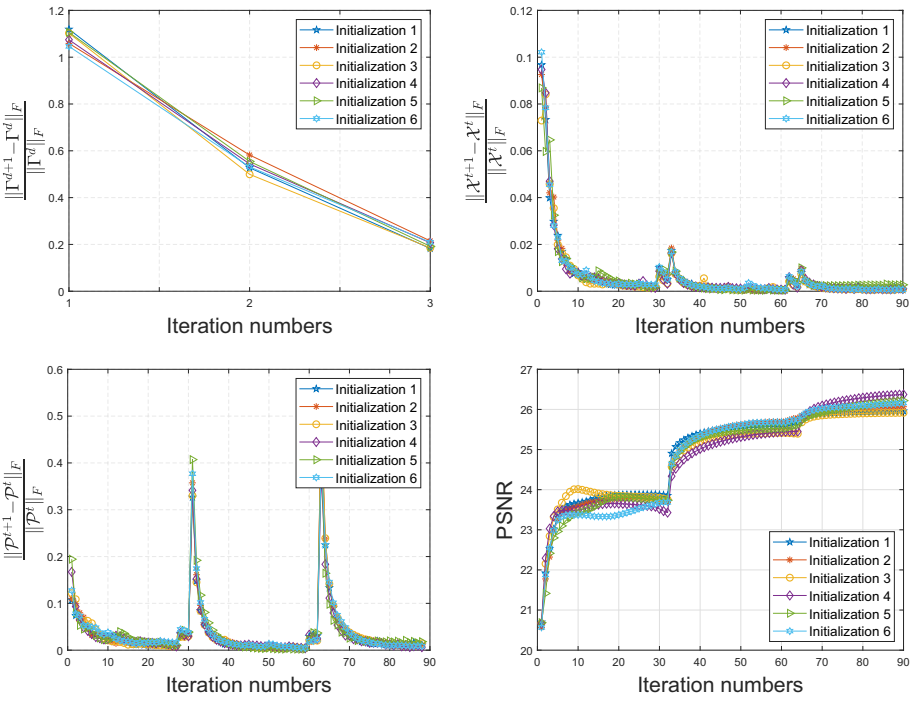

Fig. 9 Comparing iteration behaviors of DTDL in different random initializations. The relative change of Γ, $\mathcal{X} = \mathcal{L} \times_1 D_1 \times_2 D_2 \times_3 D_3$, $\mathcal{P} = [\mathcal{E}(:); \Delta\Gamma(:)]$, and PSNR are plotted

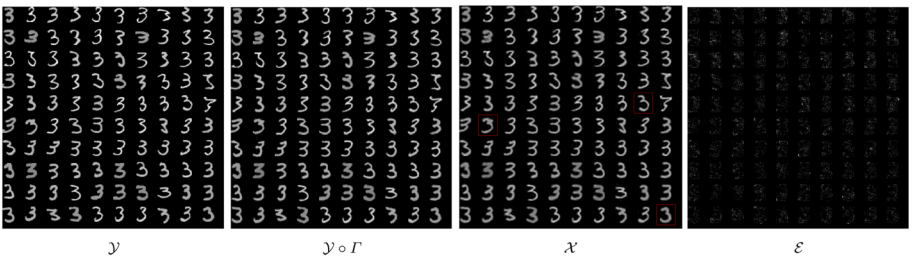

Fig. 10 Image alignment results by DTDL

outer iterations, each consisting of approximately 30 inner iterations. The first figure shows the convergence behavior of Γ during the outer iterations, with its relative change decreasing as the number of outer iterations d increases, indicating convergence. The second and third figures display the convergence behaviors of $\mathcal{X}$ and $\mathcal{P}$, respectively, with both relative changes decreasing gradually during each cycle of inner iterations within the outer iterations. The fourth figure illustrates the change in PSNR, combining inner and outer iterations, with the PSNR trend increasing throughout the iterations, indicating improved image reconstruction quality. Overall, our algorithm demonstrates consistent convergence behavior across different random initializations, highlighting its robustness.

5.3.4 Analysis of Limitations and Failure Cases

In Sect. 5.1.3, we performed image alignment for handwritten digits "3". While our method outperforms others, it still has some flaws. As observed in Fig. 10, these can be explained in two aspects: firstly, some digits are not perfectly aligned, as indicated by the red boxes in $\mathcal{X}$, although they are slightly better aligned compared to the original images; secondly, from $\mathcal{E}$, we can see that a significant number of digits still have minor details missing. Therefore, when encountering a dimension with poor correlation, directly applying our model would not yield ideal results. Instead, it is necessary to develop corresponding regularization based on the characteristics of that dimension and fine-tune the model to achieve better outcomes.

6 Conclusions and Future Work

This paper proposes a novel data-driven tensor dictionary learning (DTDL) model for image alignment, which reduces the dimensionality and complexity of the problem by factorizing the underlying third order tensor into a coefficients tensor and three dictionary matrices of smaller sizes. Moreover, the proposed model incorporates the generalized hyper-Laplacian regularization to preserve the local structures that are embedded in the underlying tensor and represented by the dictionary framework, which further enhances the alignment performance. We also develop an efficient algorithm for solving the proposed model and analyze its convergence properties. Extensive experiments on image alignment and face recognition tasks show that our method outperforms most of the state-of-the-art image alignment methods in terms of both accuracy and speed.

In real-world scenarios, high levels of noise can present substantial challenges in accurately calculating the generalized hyper-Laplacian matrix. To overcome this limitation, our future work will concentrate on developing effective preprocessing steps aimed at mitigating noise. Specifically, incorporating robust denoising techniques prior to the computation of the generalized hyper-Laplacian matrix will be explored to enhance the accuracy and reliability of the results.

A Appendix: Proof of Theorem 1

Proof Suppose that $(\mathcal{L}^1, D_i^1, \mathcal{E}^1, \Gamma^1)$ and $(\mathcal{X}^2, \mathcal{E}^2, \Gamma^2)$ are the optimal solution tuples of problems (7) and (8), respectively. Let $\mathcal{X}^1 = \mathcal{L}^1 \times_1 D_1^1 \times_2 D_2^1 \times_3 D_3^1$, we first prove that

$$\sum_{i=1}^{3} \operatorname{rank}\left(L_{(i)}^1\right) = \sum_{i=1}^{3} \operatorname{rank}\left(X_{(i)}^1\right). \tag{45}$$

Using the expression of $\mathcal{X}^1$, we have $\operatorname{rank}(X_{(i)}^1) \leq \operatorname{rank}(L_{(i)}^1)$ for $i \in [3]$. Hence,

$$\sum_{i=1}^{3} \operatorname{rank}\left(L_{(i)}^1\right) \geq \sum_{i=1}^{3} \operatorname{rank}\left(X_{(i)}^1\right). \tag{46}$$

Let $\mathcal{X}^1 = \mathcal{H}^1 \times_1 U_1^1 \times_2 U_2^1 \times_3 U_3^1$ be an orthogonal Tucker decomposition. Define $\bar{\mathcal{H}}^1$ and $\bar{U}_i^1$ with their entries as follows:

$$\bar{\mathcal{H}}^1_{j_1 j_2 j_3} = \begin{cases} \mathcal{H}^1_{j_1 j_2 j_3} & j_i \le r_i, \\ 0 & \text{otherwise}, \end{cases} \quad \bar{U}_i^1(:, j) = \begin{cases} U_i^1(:, j) & j \le r_i, \\ 0 & \text{otherwise}, \end{cases} \quad i \in [3]. \tag{47}$$

By direct computation, we have $\mathcal{X}^1 = \bar{\mathcal{H}}^1 \times_1 \bar{U}_1^1 \times_2 \bar{U}_2^1 \times_3 \bar{U}_3^1$ with

$$\operatorname{rank}\left(X^1_{(i)}\right) = \operatorname{rank}\left(\bar{H}^1_{(i)}\right) \tag{48}$$

for $i \in [3]$. Clearly, $(\bar{\mathcal{H}}^1, \bar{U}_i^1, \mathcal{E}^1, \Gamma^1)$ is a feasible solution tuple of (7), and hence

$$\sum_{i=1}^{3} \operatorname{rank}\left(L^1_{(i)}\right) + \lambda_1 R_2\left(\mathcal{E}^1\right) \le \sum_{i=1}^{3} \operatorname{rank}\left(\bar{H}^1_{(i)}\right) + \lambda_1 R_2\left(\mathcal{E}^1\right),$$

combining which with (46) and (48), we obtain (45).

From orthogonal Tucker decomposition and (47), there exist $\mathcal{L}^2 \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ and $D_i^2 \in R^{n_i \times r_i}$, $i \in [3]$ such that $\mathcal{X}^2 = \mathcal{L}^2 \times_1 D_1^2 \times_2 D_2^2 \times_3 D_3^2$ and

$$\operatorname{rank}\left(X^2_{(i)}\right) = \operatorname{rank}\left(L^2_{(i)}\right) \tag{49}$$

for $i \in [3]$. Obviously, such $(\mathcal{L}^2, D_i^2, \mathcal{E}^2, \Gamma^2)$ is a feasible solution tuple of (7). Hence we have

$$\begin{aligned} \sum_{i=1}^{3} \operatorname{rank}\left(L^1_{(i)}\right) + \lambda_1 R_2\left(\mathcal{E}^1\right) &\le \sum_{i=1}^{3} \operatorname{rank}\left(L^2_{(i)}\right) + \lambda_1 R_2\left(\mathcal{E}^2\right) \\ &= \sum_{i=1}^{3} \operatorname{rank}\left(X^2_{(i)}\right) + \lambda_1 R_2\left(\mathcal{E}^2\right), \end{aligned} \tag{50}$$

where the equality comes from (49). On the other hand, one has

$$\begin{aligned} \sum_{i=1}^{3} \operatorname{rank}\left(X^2_{(i)}\right) + \lambda_1 R_2\left(\mathcal{E}^2\right) &\le \sum_{i=1}^{3} \operatorname{rank}\left(X^1_{(i)}\right) + \lambda_1 R_2\left(\mathcal{E}^1\right) \\ &= \sum_{i=1}^{3} \operatorname{rank}\left(L^1_{(i)}\right) + \lambda_1 R_2\left(\mathcal{E}^1\right), \end{aligned} \tag{51}$$

where the inequality follows from $(\mathcal{X}^2, \mathcal{E}^2, \Gamma^2)$ being an optimal solution tuple of problem (8) and $(\mathcal{X}^1, \mathcal{E}^1, \Gamma^1)$ being a feasible solution of (8), the equality uses (45).

Hence

$$\sum_{i=1}^{3} \operatorname{rank}\left(L^1_{(i)}\right) + \lambda_1 R_2\left(\mathcal{E}^1\right) = \sum_{i=1}^{3} \operatorname{rank}\left(X^2_{(i)}\right) + \lambda_1 R_2\left(\mathcal{E}^2\right)$$

and the equivalence between problem (7) and problem (8) is established now. From the above procedure, $(\mathcal{X}^1, \mathcal{E}^1, \Gamma^1)$ is an optimal solution of (8) and $(\mathcal{L}^2, D_i^2, \mathcal{E}^2, \Gamma^2)$ is an optimal solution tuple of (7). This completes the proof. □

B Appendix: Proof of Theorem 2

Proof Suppose that $(\mathcal{L}^1, D_i^1, \mathcal{E}^1, \Gamma^1)$ and $(\mathcal{X}^2, \mathcal{E}^2, \Gamma^2)$ are the optimal solution tuples of problems (7) and (9), respectively. Let $\mathcal{X}^1 = \mathcal{L}^1 \times_1 D_1^1 \times_2 D_2^1$, we first prove that

$$\operatorname{rank}_t\left(\mathcal{L}^1\right) = \operatorname{rank}_t\left(\mathcal{X}^1\right). \tag{52}$$

It is known that [37, Lemma 2] there exists $\mathcal{D}_1^1$ and $\mathcal{D}_2^1$ such that $\mathcal{X}^1 = \mathcal{D}_1^1 * \mathcal{L}^1 * \mathcal{D}_2^1$, which together with [45, Lemma 2] implies that $\operatorname{rank}_t(\mathcal{L}^1) \geq \operatorname{rank}_t(\mathcal{X}^1)$. Let $\mathcal{X}^1 = \mathcal{H}^1 \times_1 U_1^1 \times_2 U_2^1$ be an orthogonal Tucker2 decomposition. Recalling [37, Lemma 2] again, we obtain $\mathcal{X}^1 = \mathcal{U}_1^1 * \mathcal{H}^1 * \mathcal{U}_2^1$ with

$$\begin{aligned}
&\left(U_1^1\right)^{(1)} = U_1^1, \ \left(U_1^1\right)^{(2)} = 0, \ \ldots, \ \left(U_1^1\right)^{(n_3)} = 0,\\
&\left(U_2^1\right)^{(1)} = \left(U_2^1\right)^T, \ \left(U_2^1\right)^{(2)} = 0, \ \ldots, \ \left(U_2^1\right)^{(n_3)} = 0.
\end{aligned}$$

It is clear to see that $(\mathcal{U}_1^1)^H * \mathcal{U}_1^1 = \mathcal{I}$ and $\mathcal{U}_2^1 * (\mathcal{U}_2^1)^H = \mathcal{I}$, which leads to $\mathcal{H}^1 = (\mathcal{U}_1^1)^H * \mathcal{X}^1 * (\mathcal{U}_2^1)^H$. Recalling [45, Lemma 2] again, we obtain $\operatorname{rank}_t(\mathcal{H}^1) = \operatorname{rank}_t(\mathcal{X}^1)$. Define $\bar{\mathcal{H}}^1$ and $\bar{U}_i^1$ with their entries as follows:

$$\bar{\mathcal{H}}^1_{j_1 j_2 j_3} = \begin{cases} \mathcal{H}^1_{j_1 j_2 j_3} & j_i \leq r_i, \\ 0 & \text{otherwise}, \end{cases} \quad \bar{U}_i^1(:, j) = \begin{cases} U_i^1(:, j) & j \leq r_i, \\ 0 & \text{otherwise}, \end{cases} \quad i \in [2]. \tag{53}$$

By direct computation, one has $\mathcal{X}^1 = \bar{\mathcal{H}}^1 \times_1 \bar{U}_1^1 \times_2 \bar{U}_2^1$ and $\operatorname{rank}_t(\mathcal{H}^1) = \operatorname{rank}_t(\bar{\mathcal{H}}^1)$. Thus, we have $\operatorname{rank}_t(\mathcal{X}^1) = \operatorname{rank}_t(\bar{\mathcal{H}}^1)$. Clearly, $(\bar{\mathcal{H}}^1, \bar{U}_i^1, \bar{\mathcal{E}}^1, \Gamma^1)$ is a feasible solution tuple of (7), and hence

$$\operatorname{rank}_t\left(\mathcal{L}^1\right) + \lambda_1 R_2\left(\mathcal{E}^1\right) \leq \operatorname{rank}_t\left(\bar{\mathcal{H}}^1\right) + \lambda_1 R_2\left(\bar{\mathcal{E}}^1\right) = \operatorname{rank}_t\left(\mathcal{X}^1\right) + \lambda_1 R_2\left(\mathcal{E}^1\right).$$

Therefore, the above analysis ensures $\operatorname{rank}_t(\mathcal{L}^1) = \operatorname{rank}_t(\mathcal{X}^1)$.

Then, using similar ways in the proof for Theorem 1 above, we will complete the proof of this statement. □

C Appendix: Proximal mapping

For a given proper and lower semicontinuous function $\psi : \mathfrak{L} \to [-\infty, +\infty]$, the proximal mapping associated with ψ at y is defined by

$$\operatorname{Prox}_{\lambda\psi}(y) = \arg\min_{x \in \mathfrak{L}} \ \lambda\psi(x) + \frac{1}{2}\|x - y\|_F^2.$$

The specific expressions of the proximal mappings for various instances of the function ψ are provided below.

– ℓ_q $(0 < q < 1)$: the proximal mapping of ψ is given by [14]

$$\operatorname{Prox}_{\lambda\psi}(y) = \begin{cases} 0, & \text{if } |y| < \varphi_2, \\ \{0, \operatorname{sign}(y)\varphi_1\}, & \text{if } |y| = \varphi_2, \\ \operatorname{sign}(y)z^*, & \text{if } |y| > \varphi_2, \end{cases}$$

where $\varphi_1 = [2\lambda(1-q)]^{1/(2-q)}$, $\varphi_2 = \varphi_1 + \lambda q \varphi_1^{q-1}$, $z^* \in (\varphi_1, |y|)$ is the solution of the equation $h(z) = \lambda q z^{q-1} + z - |y| = 0$ with $z > 0$.

– ℓ_1: the proximal mapping of ψ is given by [26]

$$\text{prox}_{\lambda\psi}(y) := \text{sign}(y)\max\{|y| - \lambda, 0\}.$$

– MCP: for any $0 < \lambda < \eta$, the proximal mapping of ψ is given by [39]

$$\text{Prox}_{\lambda\psi}(y) = \begin{cases} 0, & \text{if } |y| \leqslant c\lambda, \\ \frac{\text{sign}(y)(|y|-c\lambda)}{1-\lambda/\eta}, & \text{if } c\lambda < |y| \leqslant c\eta, \\ y, & \text{if } |y| > c\eta. \end{cases}$$

– SCAD: for any $0 < \lambda < b - 1$, the proximal mapping of ψ is given by [5]

$$\text{Prox}_{\lambda\psi}(y) = \begin{cases} \text{sign}(y)\max\{|y| - \xi\lambda, 0\}, & \text{if } |y| \leqslant (1+\lambda)\xi, \\ \frac{(b-1)y - \text{sign}(y)b\lambda\xi}{b-1-\lambda}, & \text{if } (1+\lambda)\xi \leqslant |y| < b\xi, \\ y, & \text{if } |y| \geqslant b\xi. \end{cases}$$

Below, we present two lemmas that provide the explicit forms of the proximal operators used in this paper.

Lemma 3 *Let $\mathcal{Z} \in \mathbb{R}^{n_1\times n_2\times n_3}$ be a given tensor. The solution $\mathcal{X}^\star$ to the optimization problem*

$$\text{Prox}_{\lambda\|\cdot\|_1^\psi}(\mathcal{Z}) := \min_{\mathcal{X}} \ \lambda\|\mathcal{X}\|_1^\psi + \frac{1}{2}\|\mathcal{X} - \mathcal{Z}\|_F^2$$

is characterized by $\mathcal{X}^\star_{ijk} = \text{Prox}_{\lambda\psi}(\mathcal{Z}_{ijk})$ for $i \in [n_1]$, $j \in [n_2]$, and $k \in [n_3]$.

Lemma 4 [20] *Let $\mathcal{Z} \in \mathbb{R}^{n_1\times n_2\times n_3}$ be a given tensor with a t-SVD decomposition $\mathcal{Z} = \mathcal{U} * \mathcal{F} * \mathcal{V}^T$. The solution $\mathcal{X}^\star$ to the optimization problem*

$$\text{Prox}_{\lambda\|\cdot\|_\circledast^\psi}(\mathcal{Z}) := \min_{\mathcal{X}} \ \lambda\|\mathcal{X}\|_\circledast^\psi + \frac{1}{2}\|\mathcal{X} - \mathcal{Z}\|_F^2$$

is characterized by $\mathcal{X}^\star = \mathcal{U} * \mathcal{D} * \mathcal{V}^T$, where $\mathcal{D}$ is an f-diagonal tensor with entries $\bar{D}^{(k)}_{i,i} = \text{Prox}_{\lambda\psi}\left(\bar{F}^{(k)}_{i,i}\right)$ for $i \in \min\{n_1, n_2\}$ and $k \in [n_3]$.

Funding Q. Yu: This author is supported by the Postgraduate Scientific Research Innovation Project of Hunan Province under project No. CX20240363. M. Bai: This author is supported by the National Natural Science Foundation of China under projects No. 11971159 and No. 12071399, and the Hunan Provincial Key Laboratory of Intelligent Information Processing and Applied Mathematics.

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Burke, J.V., Ferris, M.C.: A Gauss-Newton method for convex composite optimization. Math. Program. **71**(2), 179–194 (1995). https://doi.org/10.1007/bf01585997
2. Chen, X., Han, Z., Wang, Y., Tang, Y., Yu, H.: Nonconvex plus quadratic penalized low-rank and sparse decomposition for noisy image alignment. Sci. China Inf. Sci. **59**(5), 052107 (2016). https://doi.org/10.1007/s11432-015-5419-2

3. Cox, M., Sridharan, S., Lucey, S., Cohn, J.: Least squares congealing for unsupervised alignment of images. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2008). https://doi.org/10.1109/cvpr.2008.4587573
4. Cox, M., Sridharan, S., Lucey, S., Cohn, J.: Least-squares congealing for large numbers of images. In: 2009 IEEE 12th International Conference on Computer Vision. IEEE (2009). https://doi.org/10.1109/iccv.2009.5459430
5. Fan, J., Li, R.: Variable selection via nonconcave penalized likelihood and its oracle properties. J. Am. Stat. Assoc. **96**(456), 1348–1360 (2001). https://doi.org/10.1198/016214501753382273
6. He, J., Zhang, D., Balzano, L., Tao, T.: Iterative Grassmannian optimization for robust image alignment. Image Vis. Comput. **32**(10), 800–813 (2014). https://doi.org/10.1016/j.imavis.2014.02.015
7. Kilmer, M.E., Braman, K., Hao, N., Hoover, R.C.: Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging. SIAM J. Matrix Anal. Appl. **34**(1), 148–172 (2013). https://doi.org/10.1137/110837711
8. Kilmer, M.E., Martin, C.D.: Factorization strategies for third-order tensors. Linear Algeb. Appl. **435**(3), 641–658 (2011). https://doi.org/10.1016/j.laa.2010.09.020
9. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009). https://doi.org/10.1137/07070111x
10. Learned-Miller, E.: Data driven image models through continuous joint alignment. IEEE Trans. Pattern Anal. Mach. Intell. **28**(2), 236–250 (2006). https://doi.org/10.1109/tpami.2006.34
11. Li, M., Li, W., Xiao, M.: Nonconvex multi-view subspace clustering via simultaneously learning the representation tensor and affinity matrix. Inverse Prob. **38**(10), 105008 (2022). https://doi.org/10.1088/1361-6420/ac8ac5
12. Liang, J., Pang, T., Liu, W., Li, X., Huang, L., Gong, X., Diao, X.: Comparison of six machine learning methods for differentiating benign and malignant thyroid nodules using ultrasonographic characteristics. BMC Med. Imaging **23**(1), 154 (2023). https://doi.org/10.1186/s12880-023-01117-z
13. Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 208–220 (2013). https://doi.org/10.1109/tpami.2012.39
14. Marjanovic, G., Solo, V.: On l_q optimization and matrix completion. IEEE Trans. Signal Process. **60**(11), 5714–5724 (2012). https://doi.org/10.1109/tsp.2012.2212015
15. Martinez, A., Benavente, R.: The AR Face Database: CVC Technical Report, 24 (1998)
16. Peng, Y., Ganesh, A., Wright, J., Xu, W., Ma, Y.: RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2233–2246 (2012). https://doi.org/10.1109/tpami.2011.282
17. Peng, Y., Meng, D., Xu, Z., Gao, C., Yang, Y., Zhang, B.: Decomposable nonlocal tensor dictionary learning for multispectral image denoising. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2014). https://doi.org/10.1109/cvpr.2014.377
18. Penney, G., Weese, J., Little, J., Desmedt, P., Hill, D., hawkes, D.: A comparison of similarity measures for use in 2-D-3-D medical image registration. IEEE Trans. Med. Imaging **17**(4), 586–595 (1998). https://doi.org/10.1109/42.730403
19. Qi, N., Shi, Y., Sun, X., Yin, B.: TenSR: Multi-dimensional tensor sparse representation. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2016). https://doi.org/10.1109/cvpr.2016.637
20. Qiu, D., Bai, M., Ng, M.K., Zhang, X.: Nonlocal robust tensor recovery with nonconvex regularization. Inverse Prob. **37**(3), 035001 (2021). https://doi.org/10.1088/1361-6420/abd85b
21. Qiu, D., Bai, M., Ng, M.K., Zhang, X.: Robust low transformed multi-rank tensor methods for image alignment. J. Sci. Comput. **87**(1), 24 (2021). https://doi.org/10.1007/s10915-021-01437-8
22. Quan, Y., Huang, Y., Ji, H.: Dynamic texture recognition via orthogonal tensor dictionary learning. In: 2015 IEEE International Conference on Computer Vision (ICCV). IEEE (2015). https://doi.org/10.1109/iccv.2015.17
23. Semerci, O., Hao, N., Kilmer, M.E., Miller, E.L.: Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. IEEE Trans. Image Process. **23**(4), 1678–1693 (2014). https://doi.org/10.1109/tip.2014.2305840
24. Sun, S., Liu, J., Chen, X., Li, W., Li, H.: Hyperspectral anomaly detection with tensor average rank and piecewise smoothness constraints. IEEE Trans. Neural Netw. Learn. Syst. **34**(11), 8679–8692 (2023). https://doi.org/10.1109/tnnls.2022.3152252
25. Tian, X., Xie, K., Zhang, H.: A low-rank tensor decomposition model with factors prior and total variation for impulsive noise removal. IEEE Trans. Image Process. **31**, 4776–4789 (2022). https://doi.org/10.1109/tip.2022.3169694
26. Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. R. Stat. Soc. Ser. B Stat Methodol. **58**(1), 267–288 (1996). https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

27. Wang, M., Wang, Q., Hong, D., Roy, S.K., Chanussot, J.: Learning tensor low-rank representation for hyperspectral anomaly detection. IEEE Trans. Cybern. **53**(1), 679–691 (2023). https://doi.org/10.1109/tcyb.2022.3175771
28. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004). https://doi.org/10.1109/tip.2003.819861
29. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 210–227 (2009). https://doi.org/10.1109/tpami.2008.79
30. Xia, S., Qiu, D., Zhang, X.: Tensor factorization via transformed tensor-tensor product for image alignment. Numer. Algorithms **95**(3), 1251–1289 (2024). https://doi.org/10.1007/s11075-023-01607-9
31. Xie, Y., Zhang, W., Qu, Y., Dai, L., Tao, D.: Hyper-Laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning. IEEE Trans. Cybern. **50**(2), 572–586 (2020). https://doi.org/10.1109/tcyb.2018.2869789
32. Xue, J., Zhao, Y., Bu, Y., Chan, J.C.W., Kong, S.G.: When Laplacian scale mixture meets three-layer transform: a parametric tensor sparsity for tensor completion. IEEE Trans. Cybern. **52**(12), 13887–13901 (2022). https://doi.org/10.1109/tcyb.2021.3140148
33. Yin, M., Gao, J., Lin, Z.: Laplacian regularized low-rank representation and its applications. IEEE Trans. Pattern Anal. Mach. Intell. **38**(3), 504–517 (2016). https://doi.org/10.1109/tpami.2015.2462360
34. Yu, Q., Bai, M.: Generalized nonconvex hyperspectral anomaly detection via background representation learning with dictionary constraint. SIAM J. Imag. Sci. **17**(2), 917–950 (2024). https://doi.org/10.1137/23m157363x
35. Yu, Q., Zhang, X.: T-product factorization based method for matrix and tensor completion problems. Comput. Optim. Appl. **84**(3), 761–788 (2023). https://doi.org/10.1007/s10589-022-00439-y
36. Yu, Q., Zhang, X., Chen, Y., Qi, L.: Low Tucker rank tensor completion using a symmetric block coordinate descent method. Numer. Linear Algeb. Appl. **30**(3), e2464 (2023). https://doi.org/10.1002/nla.2464
37. Yu, Q., Zhang, X., Huang, Z.H.: Tensor factorization-based method for tensor completion with spatio-temporal characterization. J. Optim. Theory Appl. **199**(1), 337–362 (2023). https://doi.org/10.1007/s10957-023-02287-0
38. Zeng, W.J., So, H.C.: Outlier-robust matrix completion via ℓ_p-minimization. IEEE Trans. Signal Process. **66**(5), 1125–1140 (2018). https://doi.org/10.1109/tsp.2017.2784361
39. Zhang, C.H.: Nearly unbiased variable selection under minimax concave penalty. Ann. Stat. **38**(2), 894–942 (2010). https://doi.org/10.1214/09-aos729
40. Zhang, X., Ma, X., Huyan, N., Gu, J., Tang, X., Jiao, L.: Spectral-difference low-rank representation learning for hyperspectral anomaly detection. IEEE Trans. Geosci. Remote Sens. **59**(12), 10364–10377 (2021). https://doi.org/10.1109/tgrs.2020.3046727
41. Zhang, X., Wang, D., Zhou, Z., Ma, Y.: Robust low-rank tensor recovery with rectification and alignment. IEEE Trans. Pattern Anal. Mach. Intell. **43**(1), 238–255 (2021). https://doi.org/10.1109/tpami.2019.2929043
42. Zhang, Z., Aeron, S.: Denoising and completion of 3D data via multidimensional dictionary learning. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16, 2371–2377. AAAI Press, New York, New York, USA (2016)
43. Zhao, Y.P., Li, H., Chen, Y., Wang, Z., Li, X.: Hyperspectral anomaly detection via structured sparsity plus enhanced low-rankness. IEEE Trans. Geosci. Remote Sensing **61**, 1–15 (2023). https://doi.org/10.1109/tgrs.2023.3285269
44. Zheng, M., Bu, J., Chen, C., Wang, C., Zhang, L., Qiu, G., Cai, D.: Graph regularized sparse coding for image representation. IEEE Trans. Image Process. **20**(5), 1327–1336 (2011). https://doi.org/10.1109/tip.2010.2090535
45. Zhou, P., Lu, C., Lin, Z., Zhang, C.: Tensor factorization for low-rank tensor completion. IEEE Trans. Image Process. **27**(3), 1152–1163 (2018). https://doi.org/10.1109/tip.2017.2762595