Robust Low-Rank Tensor Recovery with Rectification and Alignment

Xiaoqin Zhang, Di Wang, Zhengyuan Zhou, Yi Ma

Abstract—Low-rank tensor recovery in the presence of sparse but arbitrary errors is an important problem with many practical applications. In this work, we propose a general framework that recovers low-rank tensors, in which the data can be deformed by some unknown transformation and corrupted by arbitrary sparse errors. We present a unified presentation of the surrogate-based formulations that incorporate the feacture of rectification and alignment simultaneously, and establish worst-case error bounds of the recovered tensor. In this context, the state-of-the-art methods "RASL" and "TILT" can be viewed as two special cases of our work, and yet each only performs part of the function of our method. Subsequently, we study the optimization aspects of the problem in detail by deriving two algorithms, one based on ADMM and the other based on proximal gradient. We provide global optimality convergence guarantees for the latter algorithm, and demonstrate the performance of the former through in-depth simulations. Finally, we present extensive experimental results on public datasets to demonstrate the efficacy of our proposed framework and algorithms.

Index Terms—Low-rank tensor recovery, rectification, alignment, ADMM, proximal gradient

1 Introduction

Recent years have witnessed tremendous advances in sensorial and information technology, where massive amounts of high-dimensional data, often un-labelled, became available. A key category therein is visual data, which is typically collected by various smart imaging devices (e.g. mobile phones, cameras, surveillance and medical imaging equipment). However, such data in its raw form cannot be directly used, as it often suffers from various degradation factors, such as noise pollution [1], [2], missing observations [3], [4], partial occlusion [5], [6], misalignments [7] and so on. As such, it has become an increasingly pressing challenge to develop efficient and effective computational tools that can automatically extract the hidden structures and hence useful information from such data, which are useful for various computer vision tasks.

In the past decade, many revolutionary new tools [1], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] have been developed that enable people to recover low-dimensional structures in the form of sparse vectors or low-rank matrices in high dimensional data. Specifically, in the midst of extensive literature on matrix recovery type of problems, it has been shown that if the data is a deformed or corrupted version of an intrinsically low-rank matrix, one can recover the rectified low-rank structure despite different types of deformation (linear or nonlinear) and severe corruptions. Such concepts and methods have been successfully applied to rectify the so-called low-rank textures [19] and align multiple correlated images (such as video frames or human faces) [7], [20], [21], [22], [23].

Nevertheless, instead of matrices, much of the visual data in practical applications are given in their natural form as three-order (or even higher-order) tensors (e.g. color images, videos, hyper-spectral images, high-dynamical range images, 3D range data etc.)[24], [25], [26], where important structures or useful information will be lost if we process them as a 1-D signal or a 2D matrix. These data are often subject to all types of geometric deformation or corruptions due to change of viewpoints, illuminations or occlusions. The true intrinsic structures of the data will not be correctly or fully revealed unless these nuisance factors are undone in the processing stage. Such applications naturally led to tensor recovery problems, where matrix recovery techniques are not directly applicable. This is because standard matrix recovery tools, when applied to the data of higher-order tensorial form, such as videos or 3D range data, are only able to harness one type of low-dimensional structure at a time, and not able to exploit the low-dimensional tensorial structures in the data. For instance, the previous work of TILT rectifies a low-rank textural region in a single image [19] and yet RASL aligns multiple correlated images [7]. They are highly complementary to each other: they exploit spatial and temporal linear correlation in a given sequence of images, respectively. A natural question arises: can we simultaneously harness all such low-dimensional structures in an image sequence by viewing it as a three-order tensor?

A key challenge in successfully answering the above-raised question lies in an appropriate definition of the rank of a tensor, which corresponds to the notion of intrinsic "dimension" or "degree of freedom" for the tensorial data. Traditionally, there are two definitions of tensor rank, which are based on CP (CANDECOMP/PARAFAC) decomposition [27] and Tucker decomposition [28] respectively. Similar to the definition of matrix rank, the rank of a tensor based on CP decomposition is defined as the minimum number of rank-one decomposition

X. Zhang and D. Wang are with the Institute of Intelligent System and Decision, Wenzhou University, Zhejiang 325035, China (e-mail: zhangxiaoqinnan@gmail.com, wangdi@wzu.edu.cn).

Z. Zhou is with the Department of Electrical Engineering, Stanford University, CA, USA (zyzhou@stanford.edu).

Y. Ma is with the Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA, USA (yima@eecs.berkeley.edu).

of a given tensor. However, so defined rank is a nonconvex nonsmooth function on the tensor space, and the direct minimization of this function is an NP-hard problem. An alternative definition of tensor rank is based on the so-called Tucker decomposition, which reduces to the ranks of a set of matrices unfolded from the tensor.

Due to the recent breakthroughs in the recovery of low-rank matrices [13], [14], [15], [17], [18], the latter definition has received increasing attention. Gandy *et al.* [29] adopt the sum of the ranks of the different unfolding matrices as the rank of the tensor data, which is in turn approximated by the sum of their nuclear norms. They then apply the augmented Lagrangian method to solve the tensor completion problem with Gaussian observation noise. Instead of directly adding up the ranks of the unfolding matrices, a weighted sum of the ranks of the unfolding matrices is introduced by Liu *et al.* [30] and they also proposed several optimization algorithms to estimate missing values for tensorial visual data (such as color images). In [31], three different strategies have been developed to extend the trace-norm regularization to tensors [31]: (1) treat tensor as a matrix; (2) traditional constrained optimization of low rank tensors as in [30]; (3) mixture of low-rank tensors.

All of the above work address the tensor completion problem in which the locations of the missing entries are known, and moreover, observation noise is assumed to simple Gaussian noise. However, in practice, a fraction of the tensorial entries can be arbitrarily corrupted by some large errors, and the number and the location of the corrupted entries are unknown. Most closely related to our work is the robust tensor recovery problem by Li *et al.* [32], which has extended the Robust Principal Component Analysis [1] from recovering a low-rank matrix to the tensor case, or more precisely, proposed a method to recover a low-rank tensor with sparse errors. However, a key assumption therein is that the images that form the tensor must be well aligned. But this is not the case in many different computer vision applications: images of the same object or scene can appear drastically different even under moderate changes in the object's position or pose with respect to the camera. Furthermore, the above low-rank models break down even if the images are just slightly misaligned with respect to each other. A second issue is that the optimization algorithm presented in [32] is neither computationally efficient nor accurate. In addition, global convergence guarantee is not known for the method presented therein. Motivated by the above concerns, we present in this paper a general robust tensor recovery framework that addresses these issues, thereby greatly expanding the applicability of the tensor recovery framework in real-world applications.

1.1 Our Contributions

Our main contributions are three-fold.

First, we propose a robust low-rank tensor recovery framework that deals with sparse noise corruption, and simultaneously handles rectification and alignment. Specifically, the data samples in the tensor do not need to be well-aligned nor rectified, and can be arbitrarily corrupted with a small fraction of errors. This framework automatically performs rectification and alignment when applied to imagery data such as image sequences and video frames. In particular, existing work of RASL and TILT can be viewed as two special cases of our method. We present two closely related formulations, one based on ℓ_1 minimization (Section 3), the other based on ℓ_p minimization (Section 4). We note that in the matrix case, the ℓ_p minimization ($0 < p < 1$) is known to be more effective than ℓ_p minimization, because ℓ_1 minimization needs to impose much stronger incoherence condition than ℓ_p minimization in order to achieve *exact* recovery under sparse noise [33], [34], [35], [36], [37]. This conclusion makes intuitive sense because, in comparison to l_1 norm, ℓ_p based norms provide a closer surrogate to l_0 norm (for $0 < p < 1$). Of course, the downside is that ℓ_p minimization is non-convex, and does not enjoy theoretical convergence guarantees. In tensor space, ℓ_p minimization formulations have not be explored much. Here we provide such a formulation. Further, in both formulations, we provide worst-case error bounds that relate how much error in the worst case the recovered low-rank tensor can suffer (in comparison to the true low-rank matrix) in terms of the average sparse error. As we see in the experiments, the recovered tensors typically exhibit much smaller errors than the worst-case bounds.

Second, we present two optimization algorithms that solve the tensor recovery problem efficiently. Specifically, we apply two algorithmic paradigms, one based on ADMM, the other based on proximal gradient, and derive in detail the specific optimization algorithms. As explained in more in Section 5, each of the two algorithms has its own merits and drawbacks. ADMM converges faster in practice to near-optimal regions (and fluctuate around thereafter), but its global convergence cannot be guaranteed. The algorithm based on proximal gradient converges slightly slower in practice, but we establish the strong theoretical guarantee of global convergence for the algorithm. Both algorithms are more efficient and effective than the related previous work [7], [32] (and convergence guarantee in both papers are missing).

Third, we present in Section 6 several in-depth simulation and experimental results to demonstrate the efficacy of the proposed robust low-rank tensor recovery framework. The experiments are divided into two parts. In the first part, we work with synthetic data, where the true low-rank tensor and sparse error tensor are generated (and therefore known). We then apply our proposed tensor recovery framework to the data and compare the results and performance across several methods. In the second part, we work with two publicly available datasets and demonstrate the superior performance of our proposed methods over others.

2 Mathematical Preliminaries on Tensor

To avoid confusion, all the symbols used in this paper are chosen as follows: 1) use lowercase letters for scalars ($a, b, c \cdots$); 2) use bold lowercase letters for vectors ($\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \cdots$); 3) use capital letters for matrices ($A, B, C \cdots$); 4) use calligraphic letters for tensors ($\mathcal{A}, \mathcal{B}, \mathcal{C} \cdots$). In the following subsections, the tensor algebra, as well as the tensor rank are briefly introduced.

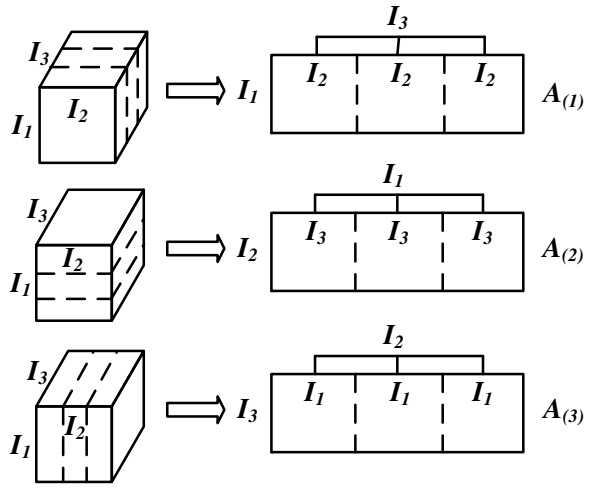


Fig. 1. Illustration of unfolding a 3-order tensor.

2.1 Tensor Algebra

A tensor can be regarded as a multi-order 'array' lying in multiple vector spaces. We denote an N-order tensor as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, where $I_n (n = 1, 2, \cdot \cdot \cdot, N)$ is a positive integer. Each element in this tensor is represented as $a_{i_1 \cdots i_n \cdots i_N}$, where $1 \leq i_n \leq I_n$. Each order of a tensor is associated with a 'mode'. By unfolding a tensor along a mode, a tensor's unfolding matrix corresponding to this mode is obtained. For example, the mode-*n* unfolding matrix $A_{(n)} \in \mathbb{R}^{I_n \times (\prod_{i \neq n} I_i)}$ of $\mathcal{A}$ consists of I_n-dimensional mode-*n* column vectors which are obtained by varying the *n*th-mode index i_n and keeping indices of the other modes fixed, represented as $\mathcal{A}_{(n)} = \text{unfold}_n(\mathcal{A})$. Fig. 1 shows an illustration of unfolding a 3-order tensor. The inverse operation of the mode-*n* unfolding is the mode-*n* folding which restores the original tensor $\mathcal{A}$ from the mode-*n* unfolding matrix $\mathcal{A}_{(n)}$, represented as $\mathcal{A} = \text{fold}_n(\mathcal{A}_{(n)})$. The mode-*n* rank r_n of $\mathcal{A}$ is defined as the rank of the mode-*n* unfolding matrix $\mathcal{A}_{(n)}$: $r_n = \text{rank}(\mathcal{A}_{(n)})$.

The operation of mode-*n* product of a tensor and a matrix forms a new tensor. The mode-*n* product of tensor $\mathcal{A}$ and matrix U is denoted as $\mathcal{A} \times_n U$. Let matrix $U \in \mathbb{R}^{I_n \times J_n}$. Then, $\mathcal{A} \times_n U \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$ and its elements are calculated by:

$$(\mathcal{A} \times_n U)_{i_1 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} = \sum_{i_n} \mathcal{A}_{i_1 \cdots i_n \cdots i_N} U_{j_n i_n} \ . \quad (1)$$

The scalar product of two tensors $\mathcal{A}$ and $\mathcal{B}$ with the same set of indices is defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} \mathcal{A}_{i_1 \cdots i_N} \mathcal{B}_{i_1 \cdots i_N} \ . \quad (2)$$

The Frobenius norm of $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as: $||\mathcal{A}||_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} = \sqrt{\sum_{i_1, \ldots, i_N} \mathcal{A}^2_{i_1 \ldots i_N}}$. Besides, denote the ℓ_0 norm $||\mathcal{A}||_0$ as the number of non-zero entities in $\mathcal{A}$ and the ℓ_1 norm $||\mathcal{A}||_1 = \sum_{i_1, \cdots, i_N} |a_{i_1, \cdots, i_N}|$ respectively. We can find that $||\mathcal{A}||_F = ||A_{(k)}||_F$, $||\mathcal{A}||_0 = ||A_{(k)}||_0$ and $||\mathcal{A}||_1 = ||A_{(k)}||_1$ for any $1 \leq k \leq N$.

2.2 Tensor Rank

Traditionally, there are two definitions of tensor rank, which are based on CP decomposition [27] and Tucker decomposition [28], respectively.

As stated in [27], in analogy to SVD, the rank of a tensor $\mathcal{A}$ can be defined as the minimum number r for decomposing the tensor into rank-one components as follows:

$$\mathcal{A} = \sum_{j=1}^{r} \lambda_j \boldsymbol{u}_j^{(1)} \circ \boldsymbol{u}_j^{(2)} \circ \cdots \circ \boldsymbol{u}_j^{(N)} = \mathcal{D} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)}, \quad (3)$$

where $\circ$ denotes outer product, $\mathcal{D} \in \mathbb{R}^{r \times r \times \cdots \times r}$ is an N-order diagonal tensor whose (j, j, j)th element is λ_j, and $U^{(n)} = [\boldsymbol{u}_1^{(n)}, \ldots, \boldsymbol{u}_r^{(n)}]$. The above decomposition model is called PARAFAC. However, this rank definition is a highly nonconvex discontinuous function on the tensor space. In general, direct minimization of such a rank function is NP-hard.

Another kind of rank definition considers the mode-*n* rank r_n of tensors, which is inspired by the Tucker decomposition [28]. The tensor $\mathcal{A}$ can be decomposed as follows:

$$\mathcal{A} = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)}, \quad (4)$$

where $\mathcal{G} = \mathcal{A} \times_1 U^{(1)^T} \times_2 U^{(2)^T} \cdots \times_N U^{(N)^T}$ is the core tensor controlling the interaction between the N mode matrices $U^{(1)}, \ldots, U^{(N)}$. In the sense of Tucker decomposition, an appropriate definition of tensor rank should satisfy the follow condition: a low-rank tensor is a low-rank matrix when unfolded appropriately. This means the rank of tensor can be represented by the rank of unfolding matrices. As illustrated in [28], the orthonormal column vectors of $U^{(n)}$ span the column space of the mode-*n* unfolding matrix $\mathcal{A}_{(n)} (1 \leq n \leq N)$, so that if $U^{(n)} \in \mathcal{R}^{I_n \times r_n}, n = 1, \ldots, N$, then the rank of the mode-*n* unfolding matrix $\mathcal{A}_{(n)}$ is r_n. Accordingly, we call $\mathcal{A}$ a rank-$(r_1, \ldots, r_N)$ tensor. We adopt this kind of definition in the following work.

From Eqs. (3) and (4), we can find that a rank-r tensor defined by the CP decomposition is always a rank-$(r, \cdots, r)$ tensor in the the sense of Tucker decomposition. The rank definition defined by Tucker decomposition is consistent with the CP decomposition, and is easy to be calculated, so we adopt this kind of definition in our work.

3 Robust Low-Rank Tensor Recovery via ℓ_1 minimization

In this section, we consider the problem of recovering low-rank tensors corrupted by sparse errors via ℓ_1 minimization. We first present the vanilla ℓ_1-minimization problem, followed by an enhanced formulation that incorporates the feature of rectification and alignment. Worst case error bound on low-rank tensor recovery is also given.

3.1 Basic Tensor Recovery Formulation

Given an N-way tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, if the tensor data $\mathcal{A}$ is pure, then we can easily extract its low rank structure. However, the data are inevitably corrupted by noise or errors

in real applications. Rather than modeling the noise with a small Gaussian, we model it with an additive sparse error term $\mathcal{E}$ which fulfills the following conditions: (1) only a small fraction of entries are corrupted; (2) the errors are large in magnitude; (3) the number and the location of the corrupted data are unknown. Mathematically, we posit that the original tensor data $\mathcal{A}$ can be decomposed into a low-rank component $\mathcal{L}_0$ and a sparse component $\mathcal{E}_0$:

$$\mathcal{A} = \mathcal{L}_0 + \mathcal{E}_0 \ . \tag{5}$$

The ultimate goal of this work is to recover (or approximately recover) the low-rank component from the erroneous observations $\mathcal{A}$.

When the corruption is modeled, the low-rank structure recovery problem for tensors can be formalized as follows.

$$\min_{\mathcal{L},\mathcal{E}} \quad \text{rank}(\mathcal{L}) + \gamma||\mathcal{E}||_0, \quad \text{s.t.} \quad \mathcal{A} = \mathcal{L} + \mathcal{E}. \tag{6}$$

The above optimization problem is not directly tractable since both rank and ℓ_0-norm are nonconvex and discontinuous. To relax this limitation, we first recall the tensor rank definition in Section 2.2. In our work, we adopt the rank definition based on the Tucker decomposition which can be represented as follows: $\mathcal{L}$ is a rank-$(r_1, r_2, \ldots, r_N)$ tensor where r_i is the rank of the unfolding matrix $L_{(i)}$. In this way, tensor rank can be converted to calculating a set of matrices' rank. We know that the nuclear (or trace) norm is the convex envelop of the rank of matrix: $||\mathcal{L}_{(i)}||_* = \sum_{k=1}^{m} \sigma_k(\mathcal{L}_{(i)})$, where $\sigma_k(\mathcal{L}_{(i)})$ is kth singular value of matrix $\mathcal{L}_{(i)}$. Therefore, we define the nuclear norm of a N-order tensor as follows:

$$||\mathcal{L}||_* = \sum_{i=1}^{N} \alpha_i ||L_{(i)}||_* \ . \tag{7}$$

We assume $\sum_{i=1}^{N} \alpha_i = 1$ to make the definition consistent with the form of matrix. The rank of $\mathcal{L}$ is replaced by $||\mathcal{L}||_*$ to make a convex relaxation of the optimization problem. Moreover, it is well known that ℓ_1-norm is a good convex surrogate of the ℓ_0-norm. We hence replace the $||\mathcal{E}||_0$ with $||\mathcal{E}||_1$ and the optimization problem in (6) becomes

$$\min_{\mathcal{L},\mathcal{E}} \quad \sum_{i=1}^{N} \alpha_i ||L_{(i)}||_* + \gamma||\mathcal{E}||_1, \quad \text{s.t.} \quad \mathcal{A} = \mathcal{L} + \mathcal{E}. \tag{8}$$

3.2 Tensor Recovery with Transformations: Simultaneous Rectification and Alignment

An explicit assumption in Eq. (8) is that it requires the tensor to be well aligned. For real data such as video and face images, the image frames (face images) should be well aligned to ensure that the (typically three-order) tensor of the image stack to have low-rank. However, for most practical data, precise alignments are not always guaranteed and even small misalignments will break the low-rank structure of the data. Without loss of generality, in this paper we focus on the 3-order tensors to study the low-rank recovery problem[1]. Most of the practical data and applications we experiment with belong to this class of tensors. To compensate possible misalignments, we introduce a set of transformations which can act on the two-dimensional slices (matrices) of the tensor data. The detail model is described as follows.

3.2.1 Transformed ℓ_1 Minimization Formulation

Consider a low-rank 3-order tensor data $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. In most visual applications, a three-order tensor can be naturally partitioned into a set of matrices (such as image frames in a video) and transformations should be applied on these matrices. In this sense, we apply a transformation to each matrix (e.g. affine or planar homography) so that the low-rank structure of tensor is guaranteed after the transformations. We denote the set of transformations $\Gamma = \{\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_{I_3}\}$, and then Eq. (8) can be changed to

$$\min_{\mathcal{L},\mathcal{E},\Gamma} \quad \sum_{i=1}^{3} \alpha_i ||L_{(i)}||_* + \gamma||\mathcal{E}||_1, \quad \text{s.t.} \quad \mathcal{A} \circ \Gamma = \mathcal{L} + \mathcal{E} \ , \tag{9}$$

where $\mathcal{A} \circ \Gamma$ means applying the transformation $\boldsymbol{\tau}_i$ to each matrix $\mathcal{A}(:,:,i), i = 1, \ldots, I_3$.

However, the equality constraint $\mathcal{A} \circ \Gamma = \mathcal{L} + \mathcal{E}$ is highly nonlinear due to the domain transformation Γ, making the program (9) not tractable. It is well known that linearization with respect to the transformation Γ parameters is a popular way to approximate the above constraint when the change in Γ is small or incremental. Accordingly, the first-order approximation to the above problem is as follows:

$$\begin{aligned} \min_{\mathcal{L},\mathcal{E},\Delta\Gamma} \quad & \sum_{i=1}^{3} \alpha_i ||L_{(i)}||_* + \gamma||\mathcal{E}||_1 \\ \text{s.t.} \quad & \mathcal{A} \circ \Gamma + \text{fold}_3 \left((\sum_{i=1}^{I_3} J_i \Delta\Gamma \boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^\top)^\top \right) = \mathcal{L} + \mathcal{E}, \end{aligned} \tag{10}$$

where J_i represents the Jacobian of $\mathcal{A}(:,:,i)$ with respect to the transformation parameters $\boldsymbol{\tau}_i$, and $\boldsymbol{\epsilon}_i$ denotes the standard basis for $\mathbb{R}^{I_3}$. As this linearization is only a local approximation to problem (9), we solve it iteratively in order to converge to a (local) minimum of (9).

3.2.2 Differences to Previous Work

As we see in Eq. (10), the optimization problem is similar to the problems addressed in [7], [19]. However, the proposed work differs from these earlier work in that:

1) RASL and TILT can be viewed as two special cases of our work. Notice in the bottom row of Fig. 1, the mode-3 unfolding matrix $A_{(3)}$. Suppose the tensor is comprised of a set of images, if we set $\alpha_1 = 0$, $\alpha_2 = 0$ and $\alpha_3 = 1$, our formulation reduces to RASL. While for the mode-1 and mode-2 unfolding matrices (see Fig. 1), if we set $\alpha_1 = 0.5$, $\alpha_2 = 0.5$ and $\alpha_3 = 0$, the proposed work acts as TILT. In this sense, our formulation is more general as it tends to simultaneously perform rectification and alignment.
2) *Our work vs. RASL:* In the image alignment applications, RASL treats each image as a vector and does not make use of any spatial structure within each image. In contrast, as shown in Fig. 1, in our work, the low-rank constraint

1. The proposed low-rank structure recovery model can be easily extended to high order (> 3) tensor data

on the mode-1 and mode-2 unfolding matrices effectively harnesses the spatial structures within images.

3) *Our work vs. TILT:* TILT deals with only one image and harnesses spatial low-rank structures to rectify the image. However, TILT ignores the temporal correlation among multiple images. While our work combines the merits of RASL and TILT, and thus can extract more structure and information in the visual data.

3.2.3 Worst-Case Error Bound

We consider a generalized version of problem (9) as follows:

$$\begin{aligned} \min_{\mathcal{L},\mathcal{E},\Gamma} \quad & ||\mathcal{L}||_* + \gamma||\mathcal{E}||_1 = \sum_{i=1}^{N} \alpha_i ||L_{(i)}||_* + \gamma||\mathcal{E}||_1 \\ \text{s.t.} \quad & \mathcal{A}\circ\Gamma = \mathcal{L} + \mathcal{E}\ . \end{aligned} \tag{11}$$

A standard way to measure the degree of recovery is to use the average recovery error on the low-rank component. Specifically, if the solution to the convex program (11) is $(\mathcal{L}^*, \mathcal{E}^*)$ and the pair of true low-rank and sparse tensors is $(\mathcal{L}_0, \mathcal{E}_0)$, then the average recovery error is defined as $\mathbf{Err}(\mathcal{L}^*) = \frac{||\mathcal{L}^* - \mathcal{L}_0||_F}{M}$, where $M = \Pi_{i=1}^{N} I_i$ is the total number of entries in the tensor. Next, we give a worst-case average recovery error bound, which quantifies how well the solution to the transformed ℓ_1 minimization problem approximates the low-rank tensor. Note that this error bound is a worst-case bound: in practice, as we demonstrate in simulations and experiments in Section 6, real average recovery error can be significantly smaller than the worst-case bound.

Theorem 1. *Let $(\mathcal{L}_0, \mathcal{E}_0)$ be the pair of true low-rank and sparse tensors and $\mathcal{L}^*$ be an optimal solution to the optimization problem (11). If the mean of the entries of the sparse component $\mathcal{E}_0$ is bounded by T, and the cardinality of the support $\mathcal{E}_0$ is bounded by m, then $\mathbf{Err}(\mathcal{L}^*) \leq \frac{2mT}{M(1-\frac{1}{\gamma}\sum_{i=1}^{N}\alpha_i\sqrt{I_i})}$ if $\gamma > (\sum_{i=1}^{N} I_i^2)^{\frac{1}{4}}$.*

By specializing the parameters $\{\alpha_i\}_{i=1}^{K}$ and γ to different values, we are able to derive a family of bounds. Next we give a particularly simple bound.

Corollary 2. *By properly choosing $\{\alpha_i\}_{i=1}^{K}$ and γ, we have $\mathbf{Err}(\mathcal{L}^*) \leq \frac{4mT}{M}$.*

Proof: Take $\alpha_i = \frac{1}{N}$,$\gamma = 2\max_i\{\sqrt{I_i}\}_{i=1}^{N}$, we have

$$\begin{aligned} ||\mathcal{L}_0 - \mathcal{L}^*||_F &= \frac{2mT}{1 - \frac{1}{2\max_i\{\sqrt{I_i}\}_{i=1}^{N}} \sum_{i=1}^{N} \frac{1}{N}\sqrt{I_i}} \\ &\leq \frac{2mT}{1 - \frac{1}{2N}\sum_{i=1}^{N}\frac{\sqrt{I_i}}{\sqrt{I_i}}} = 4mT. \end{aligned} \tag{12}$$

Note that under this choice of parameters, it holds that $1 > \frac{1}{\gamma}\sum_{i=1}^{N}\alpha_i\sqrt{I_i}$. However, $\gamma = 2\max_i\{\sqrt{I_i}\}_{i=1}^{N}$ is not necessarily larger than $(\sum_{i=1}^{N} I_i^2)^{\frac{1}{4}}$(a simple example is all I_i are equal). □

A final remark on the average recovery error bound. $\frac{m}{M}$ is the sparsity coefficient and T is the average value of all the non-zero components of the sparse error matrix. In a very sparse matrix ($\frac{m}{M} << 1$), if T is bounded (the entries in visual data are typically bounded by a constant that is not too large, i.e. the biggest value of entry is 255 for images), then the error bound is rather small, indicating rather good recovery.

4 Robust Low-Rank Tensor Recovery via ℓ_p Minimization

In this section, we extend the ℓ_1 minimization formulation to the ℓ_p minimization problem ($0 < p \leq 1$). For clarify of exposition, the development mostly parallelizes that of Section 3, albeit at a faster pace since several concepts used in this section have already been introduced in the ℓ_1 minimization case.

4.1 Transformed ℓ_p Minimization Formulation

We start by recalling a few definitions related to ℓ_p norms. First, similar to a vector, we can define the ℓ_p norm [2] for a given $\mathcal{A}$: $\|\mathcal{A}\|_{p,p} = (\sum_{i_1,\ldots,i_N}\mathcal{A}_{i_1,\ldots,i_N}^{p})^{\frac{1}{p}}$. Next, recall the Schatten-p norm $\|A\|_p$ is the ℓ_p norm on the vector of singular values: $\|A\|_p = (\sum_{i=1}^{r}\sigma_i(A)^p)^{\frac{1}{p}}$, where r is the rank of the matrix A, and $\sigma_i(A)$ is the i-th singular value of A. Following Tucker decomposition, we can similarly define Schatten-p based norm on tensors as follows: $\|\mathcal{A}\|_p = \left(\sum_{i=1}^{N}\alpha_i\|\mathcal{A}_{(i)}\|_p^p\right)^{\frac{1}{p}}$.

With the above preliminaries, we can write out the ℓ_p based tensor recovery problem, in which the nuclear norm and ℓ_1 norm in problem (9) are replaced by Schatten-p norm and ℓ_p norm respectively, and obtain the following optimization problem:

$$\min_{\mathcal{L},\mathcal{E},\Gamma} \sum_{i=1}^{3}\alpha_i||L_{(i)}||_p^p + \gamma\,\|\mathcal{E}\|_{p,p}^p, \quad \text{s.t.} \quad \mathcal{L}+\mathcal{E} = \mathcal{A}\circ\Gamma. \tag{13}$$

Linearizing around the current estimate of transformations Γ when the change in Γ is small, we again obtain the first-order approximation to problem (13) as follows:

$$\begin{aligned} \min_{\mathcal{L},\mathcal{E},\Delta\Gamma} \quad & \sum_{i=1}^{3}\alpha_i||L_{(i)}||_p^p + \gamma||\mathcal{E}||_{p,p}^p \\ \text{s.t.} \quad & \mathcal{A}\circ\Gamma + \text{fold}_3\left(\left(\sum_{i=1}^{I_3} J_i\Delta\Gamma\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_i^\top\right)^\top\right) = \mathcal{L}+\mathcal{E}, \end{aligned} \tag{14}$$

Again, as this linearization is only a local approximation to problem (13), we solve it iteratively in order to converge to a (local) minimum of (13).

4.2 Worst-Case Error Bound

Here we establish an error bound under the transformed ℓ_p minimization problem (13), which can be viewed as a generalization of the bound given in Theorem 1. To that end, we first need an auxiliary result known as the generalized power-mean inequality [38], which is stated in the following lemma.

2. Note that it is technically not a norm in our current setting where $0 < p < 1$, because the triangle inequality does not hold.

Lemma 1. *Let $w_1, w_2, \ldots, w_n$ be n positive numbers such that $\sum_{i=1}^n w_i = 1$. Then for any real numbers s, t such that $0 < s < t < \infty$, and for any $a_1, \ldots, a_n \geq 0$, it holds:*

$$(\sum_{i=1}^n w_i a_i^s)^{\frac{1}{s}} \leq (\sum_{i=1}^n w_i a_i^t)^{\frac{1}{t}}, \tag{15}$$

with equality if and only if $a_1 = a_2 = \cdots = a_n$.

We are now ready to state the worst-case error bounds:

Theorem 3. *Let $(\mathcal{L}_0, \mathcal{E}_0)$ be the pair of true low-rank and sparse tensors and $\mathcal{L}^*$ be the solution to the optimization problem (13). If the average of the entries of the sparse component $\mathcal{E}_0$ is bounded by T, and the cardinality of the support $\mathcal{E}_0$ is bounded by m, then $\boldsymbol{Err}(\mathcal{L}^*) \leq \frac{2m^{\frac{1}{p}}T}{M\sqrt[p]{1-\frac{1}{\gamma}\sum_{i=1}^N \alpha_i I_i^{1-\frac{p}{2}}}}$ if $\gamma > (\sum_{i=1}^N I_i^2)^{\frac{1}{4}}$.*

Remark 4. *Simple bounds and conditions on γ can be similarly derived as in Corollary 2, which will not be repeated here. Another thing to note is that the error bound obtained in Theorem 3 is again a worst-case bound and degenerates to the bound in Theorem 1 when $p = 1$. In the appendix, we prove this theorem for $0 < p \leq 1$ (and hence includes Theorem 1.)*

5 Optimization Algorithms

In this section, we present optimization algorithms for solving the robust tensor recovery problems with transformations in both the ℓ_1 formulation (10) and the ℓ_p formulation (14).

5.1 Optimization for ℓ_1 Minimization

We discuss the optimization aspects of the ℓ_1 minimization problem in detail. We will present two approaches for solving the ℓ_1 minimization problem, one based on ADMM and the other based on proximal gradient. Each of the two optimization algorithms has its own merits and drawbacks. At a high level, the ADMM based algorithm converges quite fast in practice, however, its global convergence is not known (and may not converge). Proximal gradient, on the other hand, converges somewhat slower than ADMM (see experiments). However, we can establish global convergence guarantees for the proximal gradient algorithm.

5.1.1 Equivalent Reformulation

Although the problem in (10) is convex, it is still difficult to solve due to the interdependent nuclear norm terms. To remove these interdependencies and optimize these terms independently, we introduce three auxiliary matrices $\{M_i, i = 1, 2, 3\}$ to replace $\{\mathcal{L}_{(i)}, i = 1, 2, 3\}$, and the optimization problem now becomes:

$$\begin{array}{ll} \min\limits_{\mathcal{L},\mathcal{E},\Delta\Gamma} & \sum\limits_{i=1}^3 \alpha_i ||M_i||_* + \gamma ||\mathcal{E}||_1 \\ \text{s.t.} & \mathcal{A} \circ \Gamma + \Delta\tilde{\Gamma} = \mathcal{L} + \mathcal{E} \\ & \mathcal{L}_{(i)} = M_i, \quad i = 1, 2, 3 \ , \end{array} \tag{16}$$

where we define $\Delta\tilde{\Gamma} \doteq \text{fold}_3\left((\sum_{i=1}^{I_3} J_i \Delta\Gamma \boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^T)^T\right)$ for simplicity. Next, we form the augmented Lagrangian function [40]:

$$\begin{aligned} f_\mu(M_i, \mathcal{L}, \mathcal{E}, \Delta\Gamma, \mathcal{Y}, Q_i) = \sum_{i=1}^3 \alpha_i ||M_i||_* + \gamma ||\mathcal{E}||_1 - \langle \mathcal{Y}, \mathcal{T} \rangle \\ + \tfrac{1}{2\mu}||\mathcal{T}||_F^2 + \textstyle\sum_{i=1}^3 \left(-\langle Q_i, O_i \rangle + \tfrac{1}{2\mu}||O_i||_F^2 \right), \end{aligned} \tag{17}$$

where we define

$$\mathcal{T} = \mathcal{L} + \mathcal{E} - \mathcal{A} \circ \Gamma - \Delta\tilde{\Gamma}, \ O_i = \mathcal{L}_{(i)} - M_i, \ i = 1, 2, 3.$$

$\mathcal{Y}$ and Q_i's are Lagrange multiplier tensors and matrices respectively. $\langle \cdot, \cdot \rangle$ denotes the inner product of matrix or tensor. μ is a positive scalar. In the following two subsections, we present optimization algorithms that solve (16): the first is ADMM-based and the second is proximal gradient based.

5.1.2 Algorithm 1: ADMM

In the augmented Lagrangian function, there are several terms that need to be optimized. To optimize these terms in a separated way, we adopt the alternating direction method of multipliers (ADMM) [41], [40], [42], which is effective to solve optimization problems with multiple terms. Per the framework of ADMM, the above optimization problem can be iteratively solved as follows.

$$\begin{cases} M_i^{k+1} : = \arg\min\limits_{M_i} f_\mu(M_i, \mathcal{L}^k, \mathcal{E}^k, \Delta\tilde{\Gamma}^k, \mathcal{Y}^k, Q_i^k); \\ \mathcal{E}^{k+1} : = \arg\min\limits_{\mathcal{E}} f_\mu(M_i^{k+1}, \mathcal{L}^k, \mathcal{E}, \Delta\tilde{\Gamma}^k, \mathcal{Y}^k, Q_i^k); \\ \mathcal{L}^{k+1} : = \arg\min\limits_{\mathcal{L}} f_\mu(M_i^{k+1}, \mathcal{L}, \mathcal{E}^{k+1}, \Delta\tilde{\Gamma}^k, \mathcal{Y}^k, Q_i^k); \\ \Delta\tilde{\Gamma}^{k+1} : = \arg\min\limits_{\Delta\tilde{\Gamma}} f_\mu(M_i^{k+1}, \mathcal{L}^{k+1}, \mathcal{E}^{k+1}, \Delta\tilde{\Gamma}, \mathcal{Y}^k, Q_i^k); \\ \mathcal{Y}^{k+1} : = \mathcal{Y}^k - \mathcal{T}^{k+1}/\mu; \\ Q_i^{k+1} : = Q_i^k - O_i^{k+1}/\mu, \quad i = 1, 2, 3. \end{cases} \tag{18}$$

In detail, we compute the solutions in analytical forms for each term is as follows.

- For term M_i $(i = 1, 2, 3)$:

$$\begin{aligned} M_i^{k+1} &= \arg\min_{M_i} \alpha_i ||M_i||_* - \langle Q_i^k, \mathcal{L}_{(i)}^k - M_i \rangle \\ &\qquad + \frac{1}{2\mu}||L_{(i)}^k - M_i||_F^2 \\ &= \arg\min_{M_i} \alpha_i ||M_i||_* + \frac{1}{2\mu}||\mathcal{L}_{(i)}^k - M_i - \mu Q_i^k||_F^2 \\ &= \arg\min_{M_i} \alpha_i \mu ||M_i||_* + \frac{1}{2}||\mathcal{L}_{(i)}^k - \mu Q_i^k - M_i||_F^2 \\ &= U_i D_{\alpha_i \mu}(\Lambda_i) V_i^\top \ , \end{aligned} \tag{19}$$

  where $U_i \Lambda_i V_i^\top = \mathcal{L}_{(i)}^k - \mu Q_i^k$ and $D_\lambda(\cdot)$ is the shrinkage operator: $D_\lambda(x) = \text{sgn}(x)\max(|x| - \lambda, 0)$[3].
- For term $\mathcal{E}$:

$$\begin{aligned} \mathcal{E}^{k+1} &= \arg\min_{\mathcal{E}} \gamma ||\mathcal{E}||_1 + \frac{1}{2\mu}||\mathcal{A} \circ \Gamma + \Delta\tilde{\Gamma}^k + \mu\mathcal{Y}^k - \mathcal{L}^k - \mathcal{E}||_F^2 \\ &= D_{\gamma\mu}\left(\mathcal{A} \circ \Gamma + \Delta\tilde{\Gamma}^k + \mu\mathcal{Y}^k - \mathcal{L}^k\right) \ . \end{aligned} \tag{20}$$

3. The extension of the shrinkage operator to vectors, matrices and tensors is applied element-wise.

- For term $\mathcal{L}$:

$$\begin{aligned}\mathcal{L}^{k+1} &= \arg\min_{\mathcal{L}} \sum_{i=1}^{3} \frac{1}{2\mu}||M_i^{k+1} + \mu Q_i^k - \mathcal{L}_{(i)}||_F^2 \\ &\quad \frac{1}{2\mu}||\mathcal{A}\circ\Gamma + \Delta\tilde{\Gamma}^k + \mu\mathcal{Y}^k - \mathcal{E}^{k+1} - \mathcal{L}||_F^2 \\ &= \frac{1}{4}\Big[\Big(\mathcal{A}\circ\Gamma + \Delta\tilde{\Gamma}^k + \mu\mathcal{Y}^k - \mathcal{E}^{k+1}\Big) \\ &\quad + \sum\nolimits_{i=1}^{3} \mathrm{fold}_i(M_i^{k+1} + \mu Q_i^k)\Big]. \end{aligned} \quad (21)$$

- For term $\Delta\tilde{\Gamma}$:

$$\begin{aligned}\Delta\tilde{\Gamma}^{k+1} &= \arg\min_{\Delta\tilde{\Gamma}} \frac{1}{2\mu}||\mathcal{A}\circ\Gamma + \Delta\tilde{\Gamma} - \mathcal{L}^{k+1} + \mu\mathcal{Y}^k - \mathcal{E}^{k+1}||_F^2 \\ &= \mathcal{L}^{k+1} + \mathcal{E}^{k+1} - \mathcal{A}\circ\Gamma - \mu\mathcal{Y}^k \ . \end{aligned} \quad (22)$$

  Here, $\Delta\tilde{\Gamma}^{k+1}$ is a tensor, and then $\Delta\Gamma^{k+1}$ can be formulated as

$$\Delta\Gamma^{k+1} = \sum_{i=1}^{n} J_i^{+}(\Delta\tilde{\Gamma}^{k+1})_{(3)}^T \boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^T \ , \quad (23)$$

  where $J_i^{+} = (J_i^T J_i)^{-1} J_i^T$ is pseudo-inverse of J_i and $(\Delta\tilde{\Gamma}^{k+1})_{(3)}$ is the mode-3 unfolding matrix of tensor $\Delta\tilde{\Gamma}^{k+1}$.

The above analytical solutions give a complete description of the ADMM-based optimization algorithm applied to the robust tensor recovery problem. Even though ADMM is effective in practice (as we shall see later in experiments), it is known that global convergence cannot be guaranteed when there are more than 2 blocks (the current optimization problem described above has 4 blocks). In the next subsection, we present a proximal gradient based optimization algorithm and establish global convergence guarantee.

5.1.3 Algorithm 2: Proximal Gradient

Applying the standard proximal gradient framework to the current problem (with appropriate regrouping) yields the following update scheme (intermediate algebraic steps are omitted):

$$\begin{cases} M_i^{k+1}: = \arg\min_{M_i} \Big\{\alpha_i||M_i||_* + \frac{1}{2\mu\tau_1}\Big\|M_i - \Big[M_i^k \\ \qquad -\tau_1(M_i^k - \mathcal{L}_{(i)}^k + \mu Q_i^k)\Big]\Big\|_F^2\Big\}; \\ \mathcal{E}^{k+1}: = \arg\min_{\mathcal{E}} \Big\{\gamma\|\mathcal{E}\|_1 + \frac{1}{2\mu\tau_1}\Big\|\mathcal{E} - \Big[\mathcal{E}^k \\ \qquad -\tau_1(\mathcal{T}^k - \mu\mathcal{Y}^k)\Big]\Big\|_F^2\Big\}; \\ \mathcal{L}^{k+1}: = \arg\min_{\mathcal{L}} \frac{1}{2\mu\tau_1}\Big\|\mathcal{L} - \Big\{\mathcal{L}^k - \tau_1\Big[\mathcal{T}^k + 3\mathcal{L}^k - \mu\mathcal{Y}^k \\ \qquad - \sum_{i=1}^{3}\mathrm{fold}_i(M_i^k + \mu Q_i^k)\Big]\Big\}\Big\|_F^2; \\ \Delta\tilde{\Gamma}^{k+1}: = \arg\min_{\Delta\tilde{\Gamma}} \frac{1}{2\mu\tau_2}\Big\|\Delta\tilde{\Gamma} - \Big[\Delta\tilde{\Gamma}^k - \tau_2(\Delta\tilde{\Gamma}^k - \mathcal{L}^{k+1} \\ \qquad -\mathcal{E}^{k+1} + \mathcal{A}\circ\Gamma + \mu\mathcal{Y}^k)\Big]\Big\|_F^2; \\ \mathcal{Y}^{k+1}: = \mathcal{Y}^k - \mathcal{T}^{k+1}/\mu; \\ Q_i^{k+1}: = Q_i^k - O_i^{k+1}/\mu, \qquad i = 1,2,3. \end{cases} \quad (24)$$

The analytical solutions for each term are given as follows.

- For term M_i^{k+1} $(i = 1, 2, 3)$:

$$M_i^{k+1} = U_i D_{\alpha_i\mu\tau_1}(\Lambda) V_i^T,$$

  where $U_i\Lambda V_i^T = M_i^k - \tau_1(M_i^k - \mathcal{L}_{(i)}^k + \mu Q_i^k)$ and $D_\lambda(\cdot)$ is the shrinkage operator: $D_\lambda(x) = \mathrm{sgn}(x)\max(|x| - \lambda, 0)$.
- For term $\mathcal{E}^{k+1}$:

$$\mathcal{E}^{k+1} = D_{\gamma\mu\tau_1}\Big(\mathcal{E}^k - \tau_1\left(\mathcal{T}^k - \mu\mathcal{Y}^k\right)\Big).$$

- For term $\mathcal{L}^{k+1}$:

$$\mathcal{L}^{k+1} = \mathcal{L}^k - \tau_1\Big[\mathcal{T}^k + 3\mathcal{L}^k - \mu\mathcal{Y}^k - \sum_{i=1}^{3}\mathrm{fold}_i(M_i^k + \mu Q_i^k)\Big].$$

- For term $\Delta\tilde{\Gamma}^{k+1}$:

$$\Delta\tilde{\Gamma}^{k+1} = \Delta\tilde{\Gamma}^k - \tau_2(\Delta\tilde{\Gamma}^k - \mathcal{L}^{k+1} - \mathcal{E}^{k+1} + \mathcal{A}\circ\Gamma + \mu\mathcal{Y}^k)$$

  We can also transform $\Delta\tilde{\Gamma}^{k+1}$ to its original form by (23).

Using proximal gradient, we next establish that the global convergence to the optimal solution can be guarantee, as indicated by the following theorem. The proof is given in the appendix.

Theorem 5. *If $0 < \tau_1 < 1/5$ and $0 < \tau_2 < 1$, then the sequence $\{M_i^k, \mathcal{L}^k, \mathcal{E}^k, \Delta\tilde{\Gamma}^k, \mathcal{Y}^k, Q_i^k,\ i = 1,2,3\}$ generated by the above proximal gradient algorithm converges to the optimal solution to Problem (10).*

5.2 Optimization for ℓ_p Minimization

First, we similarly give the equivalent reformulation of the ℓ_p minimization problem as follows:

$$\begin{aligned} \min_{\mathcal{L},\mathcal{E},\Delta\Gamma} \quad & \sum_{i=1}^{3}\alpha_i||M_i||_p^p + \gamma||\mathcal{E}||_{p,p}^p \\ \text{s.t.} \quad & \mathcal{A}\circ\Gamma + \Delta\tilde{\Gamma} = \mathcal{L} + \mathcal{E} \\ & \mathcal{L}_{(i)} = M_i, \quad i = 1,2,3 \ , \end{aligned} \quad (25)$$

As before, we can apply either ADMM or proximal gradient to solve this optimization problem. Indeed, in both algorithms, each step still admits analytical solutions (albeit different from those for ℓ_1 minimization). However, one crucial difference here is that the ℓ_1 minimization problem is not convex. Consequently, global convergence of the proximal gradient algorithm cannot be guaranteed (recall that the global convergence guarantee is the only advantage of proximal gradient over ADMM in ℓ_1 minimization). In light of this (and of space concerns), we will only present the ADMM algorithm, as it converges faster in practice compared to proximal gradient.

The structure of ADMM is quite similar, so we provide a quick presentation here, mostly focused on the differences from ℓ_1 minimization. First, we can write out the augmented Lagrangian function:

$$\begin{aligned} f_\mu(M_i, \mathcal{L}, \mathcal{E}, \Delta\Gamma, \mathcal{Y}, Q_i) &= \sum_{i=1}^{3}\alpha_i||M_i||_p^p + \gamma||\mathcal{E}||_{p,p}^p - \langle\mathcal{Y}, \mathcal{T}\rangle \\ &\quad + \tfrac{1}{2\mu}||\mathcal{T}||_F^2 + \sum\nolimits_{i=1}^{3}\Big(-\langle Q_i, O_i\rangle + \tfrac{1}{2\mu}||O_i||_F^2\Big), \end{aligned} \quad (26)$$

Then ADMM proceeds in the steps of (18). The analytical solutions for solving M_i and $\mathcal{E}$ are different from the ℓ_1 minimization case, which we give the details below:

- For term M_i^{k+1} $(i = 1, 2, 3)$:

$$M_i^{k+1} = \arg\min_{M_i} \alpha_i \mu ||M_i||_p^p + \frac{1}{2}||\mathcal{L}_{(i)}^k - \mu Q_i^k - M_i||_F^2 = U_i T_{\alpha_i \mu}(\Lambda_i) V_i^\top \ , \quad (27)$$

where $U_i \Lambda_i V_i^\top = \mathcal{L}_{(i)}^k - \mu Q_i^k$ and $T_\eta(\cdot)$ is the shrinkage operator:

$$T_\eta(z) = \begin{cases} 0 & \text{if} |z| < \kappa \\ \{0, \text{sgn}(z)\hat{a}\} & \text{if} |z| = \kappa \\ \text{sgn}(z)\hat{a}^\star & \text{if} |z| > \kappa \end{cases} \quad (28)$$

In (28), $\hat{a} = [2\eta(1-p)]^{\frac{1}{2-p}}$, $\kappa = \hat{a} + \eta p \hat{a}^{p-1}$. $\hat{a}^* \in (\hat{a}, |z|)$ is the larger solution of

$$a + \eta p a^{p-1} = |z|, \text{ where } a > 0 \quad (29)$$

which can be obtained from the iteration $a_{(t+1)} = |z| - \eta p a_{(t)}^{p-1}$ with the initial value $a_{(0)} \in (\hat{a}, |z|)$.

- For term $\mathcal{E}^{k+1}$:

$$\mathcal{E}^{k+1} = \arg\min_{\mathcal{E}} \gamma ||\mathcal{E}||_{p,p}^p + \frac{1}{2\mu}||\mathcal{A} \circ \Gamma + \Delta\tilde{\Gamma}^k + \mu\mathcal{Y}^k - \mathcal{L}^k - \mathcal{E}||_F^2 = T_{\gamma\mu}\left(\mathcal{A} \circ \Gamma + \Delta\tilde{\Gamma}^k + \mu\mathcal{Y}^k - \mathcal{L}^k\right) \quad (30)$$

6 Experimental Results

In this section, we present experiments on several synthetic and real-world datasets with the following five algorithms:

1) RASL [7] implemented by algorithm IALM (Inexact Augmented Lagrange Multiplier)[4].
2) Li's work [32].
3) ADMM with ℓ_1-norm (denoted as ℓ_1+ADMM).
4) Proximal gradient with ℓ_1-norm (denoted as ℓ_1+PG).
5) ADMM with ℓ_p-norm (denoted as ℓ_p+ADMM).

Note that the last three are the algorithms proposed in this paper (see the previous section). We choose RASL and Li's work for comparison because: (1) They represent the state-of-the-art works that address similar problems as ours. (2) The effectiveness and efficiency of our methods for recovery of low-rank tensors can be validated by comparing our work with RASL and Li's work. These algorithms are tested with several synthetic and real-world datasets, and the results are both qualitatively and quantitatively analyzed.

6.1 Synthetic results

This part tests the above five algorithms with synthetic data. To make a fair comparison, we start by clarifying some implementation details: (1) Since domain transformations are not considered in Li's work, we assume the synthetic data are well aligned. (2) RASL is implemented without transformations. (3) Since RASL is applied to one mode of the tensor, to make it more competitive, we apply RASL to each mode of the tensor and take the mode that has the minimal reconstruction error. (4) The maximum iterations and the stopping tolerance are respectively set to 500 and 10^{-8}.

For synthetic data, we first randomly generate two data tensors: (1) a pure low-rank tensor $\mathcal{L}_0 \in \mathbb{R}^{50\times50\times50}$ whose rank is (10,10,10); (2) an error tensor $\mathcal{E} \in \mathbb{R}^{50\times50\times50}$ in which only a fraction c of entries are non-zero (To ensure the error to be sparse, the maximal value of c is set to 40%) and the non-zero entries are i.i.d. uniformly in the interval $[-500, 500]$. Then the testing tensor $\mathcal{A}$ can be obtained as $\mathcal{A} = \mathcal{L}_0 + \mathcal{E}$. All the above five algorithms are applied to recover the low-rank structure of $\mathcal{A}$, which is represented as $\mathcal{L}_r$. Therefore, the reconstruction error is defined as $\text{error} = \frac{||\mathcal{L}_0 - \mathcal{L}_r||_F}{||\mathcal{L}_0||_F}$. The result of a single run is a random variable, because the data are randomly generated, so the experiment is repeated 50 times to generate statistical averages.

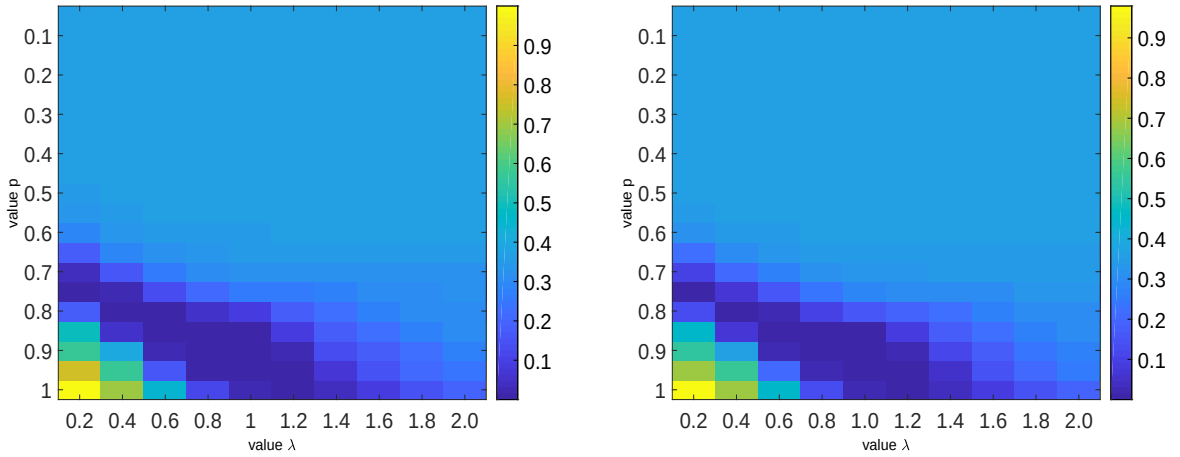
In our proposed methods, α is set to $[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ since the three modes of synthetic data has the same importance. We vary the values p, λ and c in the ranges $\{0.05, 0.10, 0.15, \cdots, 1\}$, $\{0.2, 0.4, 0.6, \cdots, 2\}$ and $\{1, 2, \ldots, 40\}$ respectively, and obtain the mean values of reconstruction errors for the optimization algorithms ADMM and proximal gradient as shown in Fig. 2. From the results, we can conclude that ℓ_p- norm for proper value p and λ is superior than ℓ_1- norm in data recovery respect, and the optimal p and λ is 0.85 and 1 respectively.

The parameters of RASL and Li's work are also selected carefully by the smallest mean values of reconstruction errors (with the same averaging procedure as described in Figure 2). The left column of Fig. 3 shows reconstruction error, from which we can see that the tensor recovery methods including our algorithms and Li's work are superior than RASL since tensor can retain more richer structures and information than matrices. The reconstruction errors of RASL, Li's work, ℓ_1+PG and ℓ_1+ADMM increase sharply with the proportion of corrupted entries increasing. ℓ_1+PG and ℓ_1+ADMM have the similar performance, and they are more powerful than the methods of RASL and Li's work. ℓ_p+ADMM achieves the most accurate result for reconstruction among all the algorithms. Even when 40% of entries are corrupted, the reconstruction error is still about 10^{-8}. As shown in the middle column of Fig. 3, comparing with Li's work, our works can achieve at least 2 times speed-up. Moveover, the result shows that the average running time of our work is higher than RASL. This is because RASL only optimizes on a single mode while our work optimize on all three modes and the variables evolved in our work are about three times of those in RASL. The above results demonstrate the effectiveness and efficiency of our proposed optimization method for low-rank tensor recovery.

6.2 Image Sequence Recovery and Alignment

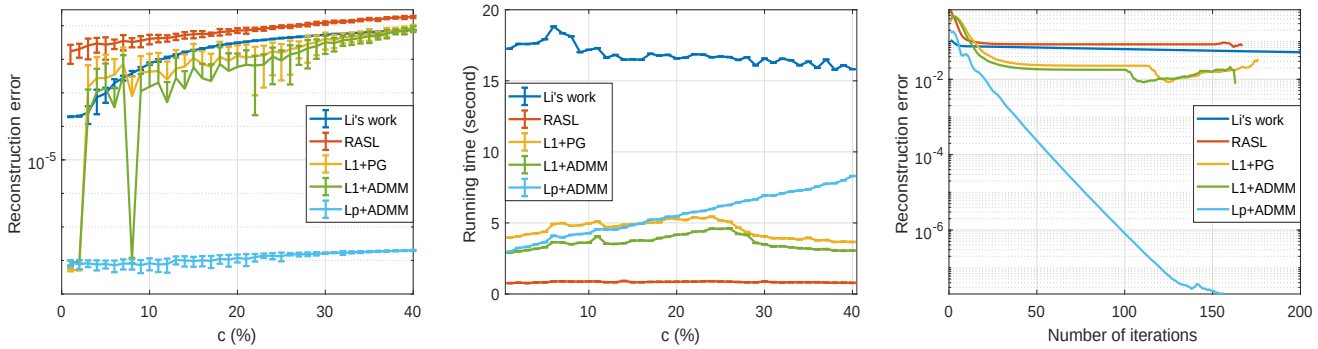
In this section, we apply all five algorithms to several real-world datasets. The first dataset contains 16 images of the side of a building, taken from various viewpoints by a perspective camera, and with various occlusions due to tree branches. The second data set contains 100 images of the handwritten number

4. For more detail, please refer to http://perception.csl.illinois.edu/matrix-rank/sample_code.html



(a) ℓ_p+ADMM (b) ℓ_p+PG

Fig. 2. The reconstruction errors with different values p and λ. For each value p, λ and c, we perform 50 experiments and obtain the averaged error. We then average over c to obtain the mean reconstruction error for each p and λ.



(a) Reconstruction errors (b) Time costs (c) Reconstruction errors with iterations

Fig. 3. Results on synthetic data.

"3", with a fair amount of diversity. For example, as shown in Fig. 4(a), the number "3" in the column 1 and row 6 is barely recognizable. The third data set contains 140 frames of a video showing Al Gore talking.

To fully demonstrate the effectiveness of our proposed algorithms, we divide this subsection further into three parts: recovery on raw images; recovery on images with salt and pepper noises; recovery on images with occlusions. Since both RASL and our algorithms perform transformation while Li's work doesn't, for fairness of comparison, we apply RASL's transformation on the image data before feeding them as inputs to Li's work.

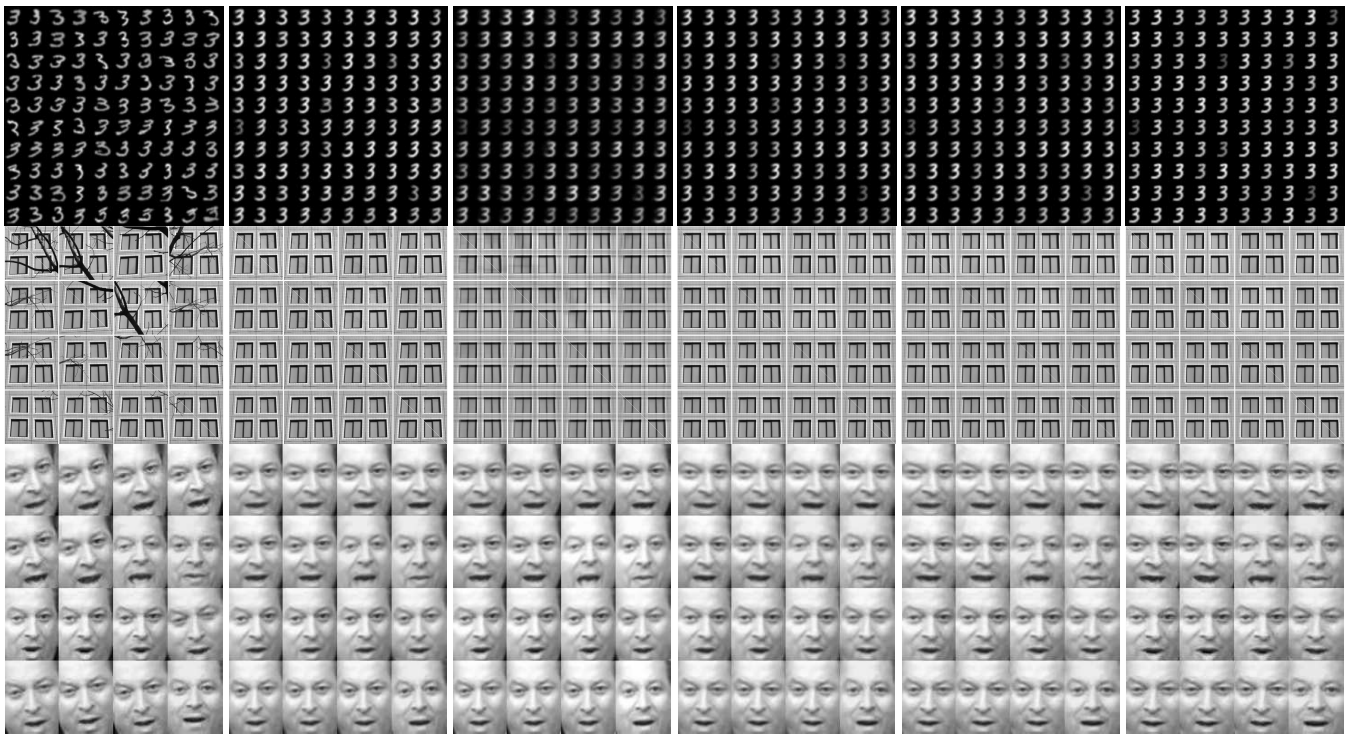
6.2.1 *Raw image*

Fig. 4 illustrates the low-rank recovery results on the three datasets, in which Fig. 4(a) shows the original image and Fig. 4(b)-(f) show the results of RASL, Li'work, ℓ_1+PG, ℓ_1+ADMM and ℓ_p+ADMM respectively. As shown in the top row of Fig. 4, We can see that our works, especially ℓ_p+ADMM, have achieved better performance than the other two algorithms from human's perception, in which the 3's are more clear and their poses are upright. The results of dataset "windows" are illustrated in the middle row of Fig. 4. It can be concluded that that our works and Li's work not only remove the branches from the windows, but also rectify window position. Moreover, the result obtained by our works is noticeably sharper than Li's work. The bottom row of Fig. 4 is the recovery results on dataset "Al Gore". We can see that the face alignment results obtained by our works are significantly better than those obtained by the other two algorithms. The reason is that human face has a rich spatial low-rank structures due to symmetry, and our methods harness both temporal and spatial low-rank structures for rectification and alignment.

6.2.2 *Image with salt and pepper noises*

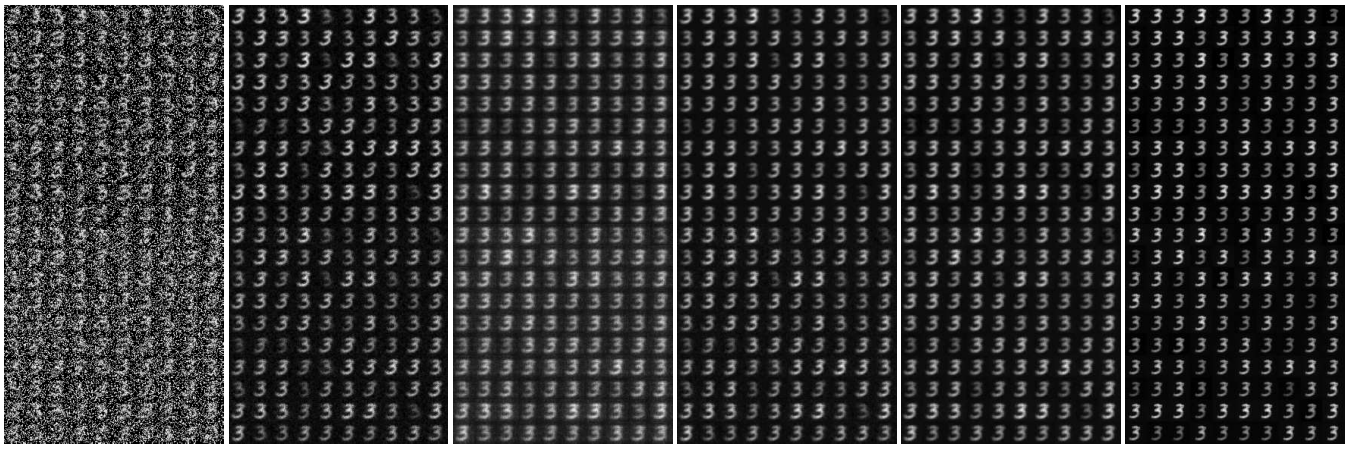
To further verify the performance of our works on image sequence with sparse noises, we add different proportions of salt and pepper noises to the original image sequence data.

In Fig. 5 - 7, column (a) shows the original images with added salt and pepper noises. In the top row and bottom row, the percents of added noises are respectively 40% and 50%. The recovery results of the five algorithms are shown in



(a) Input images (b) RASL (c) Li's work (d) ℓ_1+PG (e) ℓ_1+ADMM (f) ℓ_p+ADMM

Fig. 4. Recovery results on the data set without adding noises.



(a) Input images (b) RASL (c) Li's work (d) ℓ_1+PG (e) ℓ_1+ADMM (f) ℓ_p+ADMM

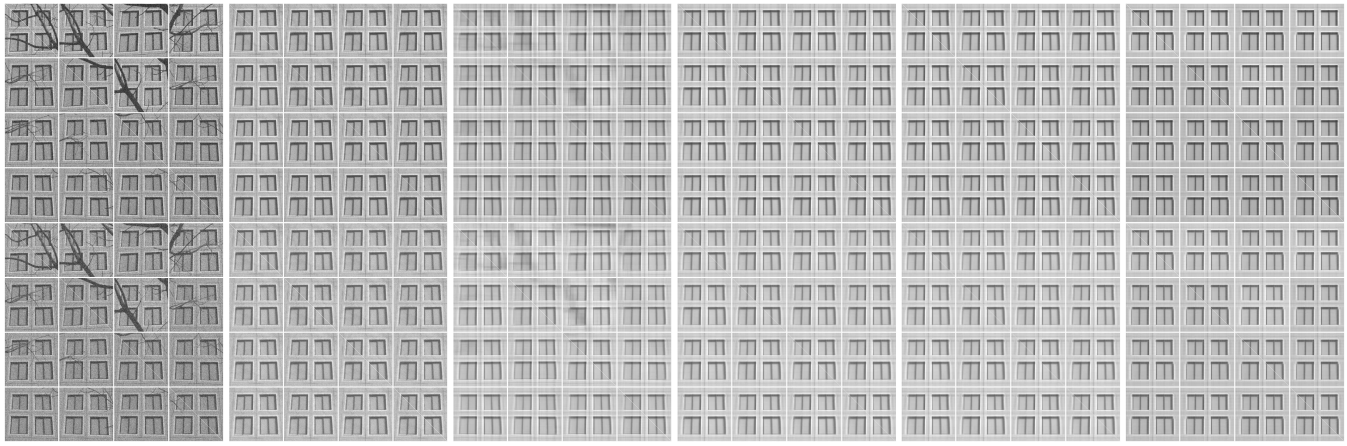
Fig. 5. Recovery results on the data set of handwritten digit "3" with different proportions of salt and pepper noises.

column (b)-(f).

From those results, we can see that although RASL removed most of the noises (still quite noisy for "Al Gore"), the images are not well-aligned. On the other hand, Li's work does not perform well in terms of removing the noises (the figure "3" and "Al Gore" are particularly blurry). Compared with RASL and Li's work, our works are more robust to the proportion of the salt and pepper noises than the other methods. Even if the proportion of noise is up to 50%, which is very difficult to recognition from human's perception (for datasets "digit 3" and "Al Gore"), our works still can efficiently recover the desired low-rank structure and have the superior performance on image alignment. In our works, ℓ_p+ADMM achieves the best results of all. The recovery images obtained by ℓ_p+ADMM is significantly shaper and more clear than the other methods. And we can see that ℓ_p norm is most effective in removing sparse noises.

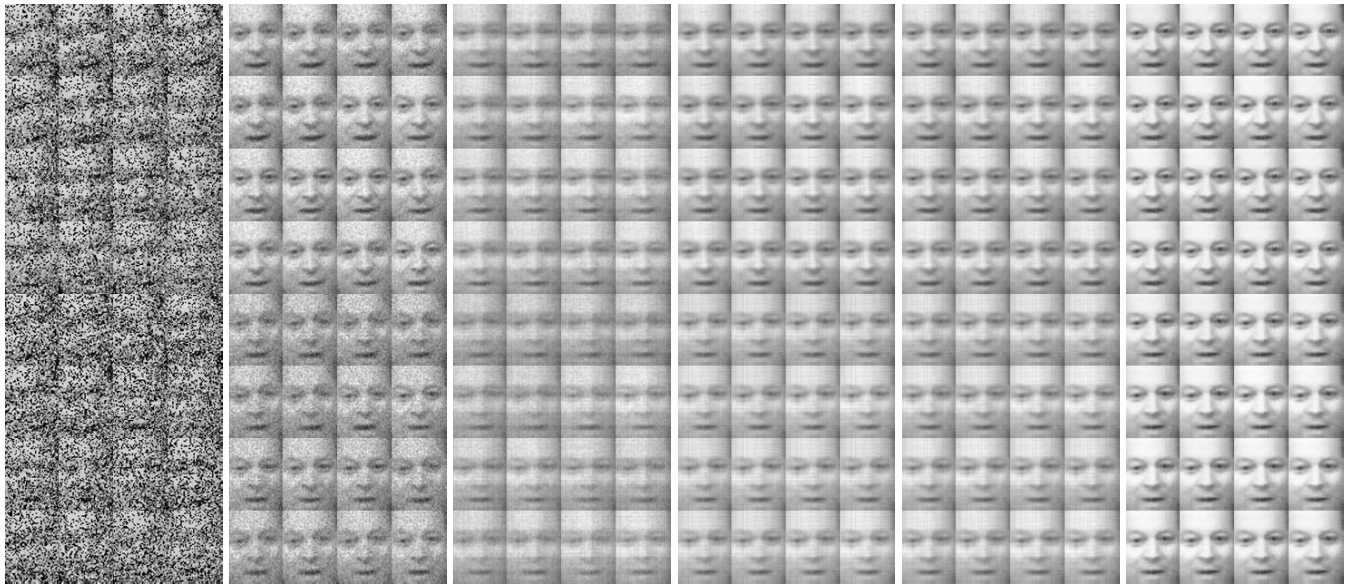
6.2.3 Image with occlusions

In this subsection, we add the image "baboon" to the image sequence data at random locations. The occluded images of the three datasets are shown in Figure 8. Note that since the "baboon" occlusions are added at random locations, this increases the difficulty level of recovery, as certain key fea-



(a) Input images (b) RASL (c) Li's work (d) ℓ_1+PG (e) ℓ_1+ADMM (f) ℓ_p+ADMM

Fig. 6. Recovery results on the data set of "Windows" with different proportions of salt and pepper noises.



(a) Input images (b) RASL (c) Li's work (d) ℓ_1+PG (e) ℓ_1+ADMM (f) ℓ_p+ADMM

Fig. 7. Recovery results on the data set of "Al Gore" with different proportions of salt and pepper noises.

tures/information of the image may be lost a result of the occlusion.

In Fig. 9-11, column (a) shows the original images with added occlusions. In the top row and bottom row, the percents of occluded area are respectively 10% and 15%. The recovery results of the five algorithms are shown in column (b)-(f). As can be seen clearly from the figure, our algorithms perform much better than the other two. In particular, ℓ_p+ADMM performs the best.

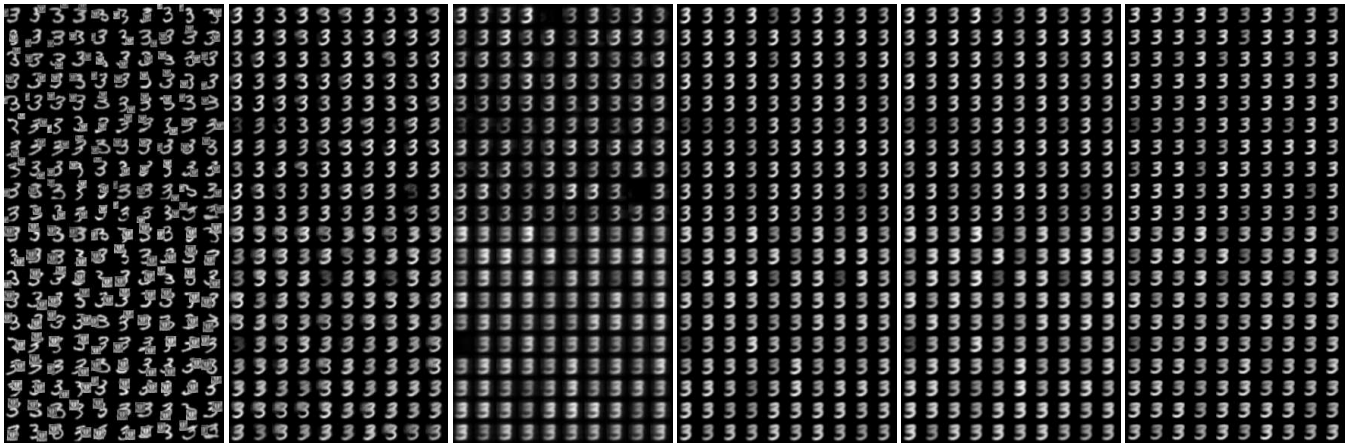
6.3 Face recognition

In this part, face recognition is conducted on the following three datasets:

- **ORL database.** The Cambridge ORL database consists 400 images for 40 different persons, and each person has 10 images. In our experiments, each images are resized to 32×32. For each person, we randomly select 5 images as training samples, and the rest are left for testing samples.
- **Extended Yale B database.** The Extended Yale B database contains $2,414$ frontal face images of 38 subjects, with approximately 64 frontal face images per subject. Each image is resized to 32×32. We randomly select 30 images as training samples for each people, and the rest are testing samples.
- **CMU PIE database.** The CMU PIE database consists more than $40,000$ facial images of 68 persons. In the experiments, we choose the first 5 subjects and 170 images per subject from varying illuminations and facial expressions, in which each image is resized into 32×32. The number of training samples and testing samples for each person are respectively 50 and 120.

(a) Digits 3 (b) Windows (c) Al Gore

Fig. 8. Images with 15 percent occlusions.



(a) Input images (b) RASL (c) Li's work (d) ℓ_1+PG (e) ℓ_1+ADMM (f) ℓ_p+ADMM

Fig. 9. Recovery results on the data set of handwritten digit "3" with different areas of occlusion.

The procedure of each experiment can be decomposed into four steps, which are described below.

Step 1 In each image, we add different proportions of salt and pepper noises, which is set to $\gamma = 0.04 : 0.04 : 0.4$.

Step 2 For each proportion of noise, we respectively use the five algorithms to recover the low-rank structure from the polluted images.

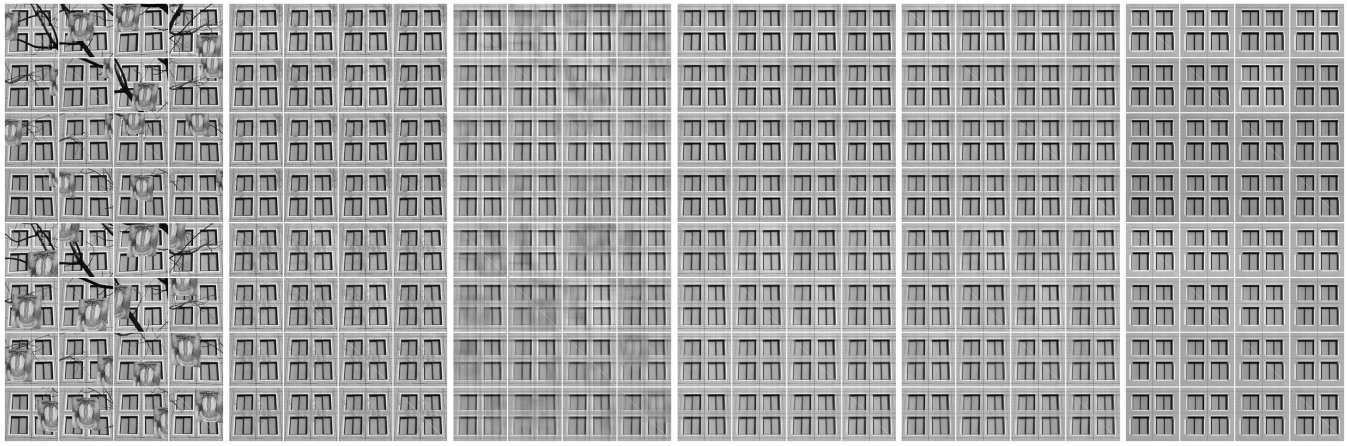
Step 3 By using the 1-nearest neighbor (1NN) algorithm to the recovered images, the recognition accuracies of the testing samples are obtained.

In our experiments, we select the optimal parameters for the five algorithms by using 10-fold cross validation. The recognition accuracies of the five algorithms on the three datasets are shown in Fig. 12. We can see that Li's work almost has the lowest accuracies, especially the noise ratio is small. The accuracies of RASL are comparable to those of our methods when the noise ratio is less than 16%. However, they have a sharp fall with the noise proportion increasing. Our works are not only superior to the other two models in terms of accuracy, but also robust to the proportion of noise (and particularly for l_p+ADMM, which has the best performance under all noise ratios).

Partial face images of 4 persons are given in Fig. 13, and we have selected 2 faces for each person. From top row to bottom row, they are respectively the original face images, the face images with noise proportion 0.32, the face images recovered by RASL, Li'work, ℓ_1+PG, ℓ_1+ADMM and ℓ_p+ADMM. It can be seen that, the images recovered by RASL and Li's work still have serious noises. ℓ_1+PG and ℓ_1+ADMM are better than RASL and Li's work, however, the recovered images are blurred. The images recovered by ℓ_p+ADMM have the best sharpness, which are close to the original face images. The good performance on face recognition shows the effectiveness of our works in image denoising.

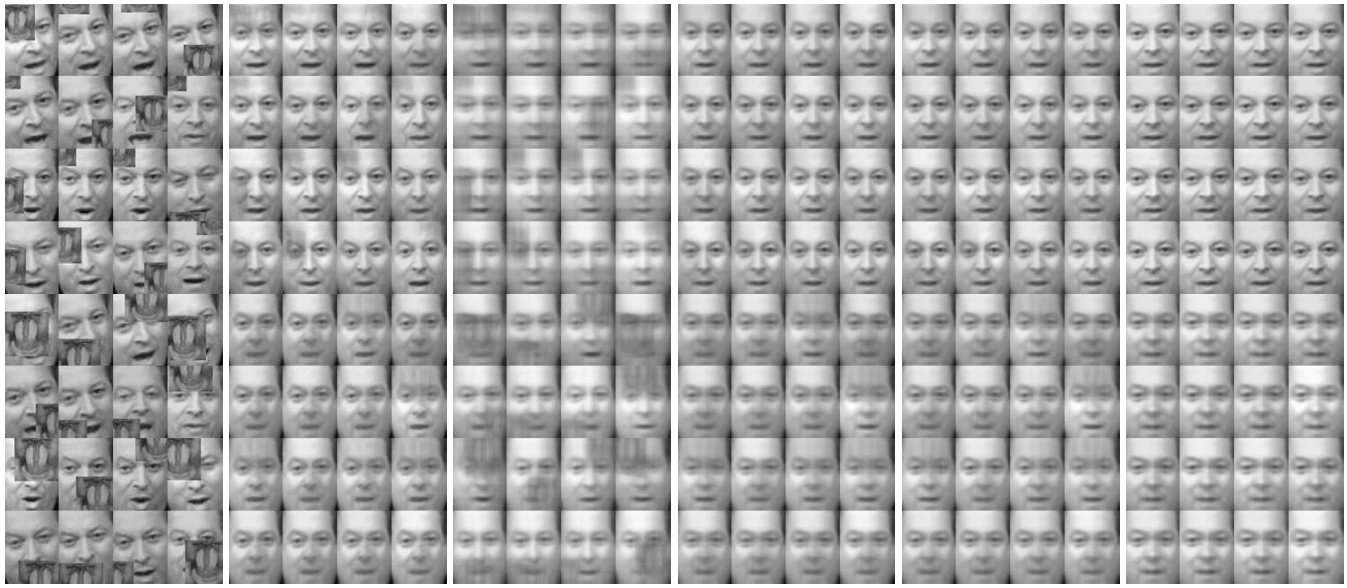
7 Conclusion

This paper has proposed a general low-rank discovery framework for arbitrary tensor data, which can simultaneously realize rectification and alignment. In the optimization process, three auxiliary variables are introduced to relax the interdependence of the nuclear norms of the unfolding matrices. A proximal gradient based alternating direction method is used for solving the optimization problem, and the convergence is guaranteed. Compared with three state-of-the-art work,



(a) Input images (b) RASL (c) Li's work (d) ℓ_1+PG (e) ℓ_1+ADMM (f) ℓ_p+ADMM

Fig. 10. Recovery results on the data set of "Windows" with different proportions of salt and pepper noises.



(a) Input images (b) RASL (c) Li's work (d) ℓ_1+PG (e) ℓ_1+ADMM (f) ℓ_p+ADMM

Fig. 11. Recovery results on the data set of "Al Gore" with different proportions of salt and pepper noises.

the correctness and effectiveness of the proposed work is validated.

REFERENCES

[1] E. Candès, X. Li, Y. Ma and J. Wright, "Robust principal component analysis?", *Journal of the ACM*, 58(3):11, 2011.

[2] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, L. Zhang, "Weighted Schatten p-norm minimization for image denoising and background subtraction, *IEEE Trans. on Image Processing*, 25(10): 4842-4857, 2016

[3] M. A. Davenport, J. Romberg, "An overview of low-rank matrix recovery from incomplete observations", *IEEE Journal of Selected Topics in Signal Processing*, 10(4): 608-622, 2016.

[4] W. Zeng, H. C. So, "Outlier-robust matrix completion via lp-minimization, *IEEE Trans. on Signal Processing*, 66(5): 1125-1140, 2018.

[5] J. Yang, L. Luo, J. Qian, Y. Tai, F. Zhang, and Y. Xu, "Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 39(1): 156-171, 2017.

[6] H. Yong, D. Meng, W. Zuo, L. Zhang, "Robust online matrix factorization for dynamic background subtraction, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2017.

[7] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: robust alignment by sparse and low-rank decomposition for linearly correlated images", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(11): 2233-2246, 2012.

[8] E. Candès and B. Recht, "Exact matrix completion via convex optimization", *Foundations of Computational mathematics*, 9(6):717-772, 2009.

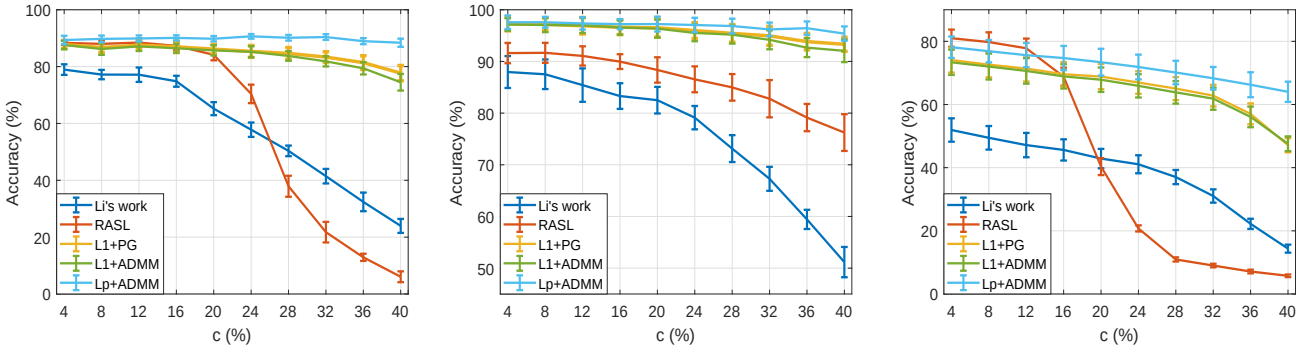
[9] J. Cai, E. Candès and Z. Shen, "A singular value thresholding algorithm for matrix completion", *SIAM Journal on Optimization*, 20(4):1956-1982, 2010.

[10] R. Keshavan, A. Montanari, S. Oh, "Matrix completion from a few entries", *IEEE Trans. on Information Theory*, 56(6):2980-2998, 2010.

[11] E. Candès and T. Tao, "The power of convex relaxation: near-optimal matrix completion", *IEEE Trans. on Information Theory*, 56(5):2053-2080, 2010.

[12] E. Candès and Y. Plan, "Matrix completion with Noise", *Proceedings of the IEEE*, 98(6):925-936, 2010.

[13] B. Recht, M. Fazel and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization", *SIAM review*, 52(3): 471-501, 2010.



(a) ORL (b) PIE5 (c) YaleB

Fig. 12. The contrast results on the three face datasets.

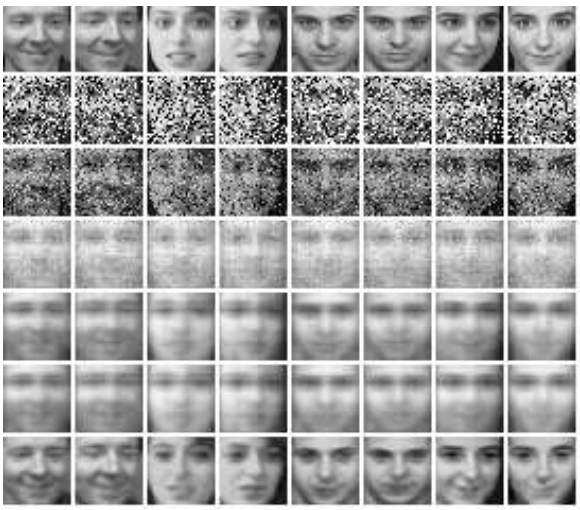


Fig. 13. Some recovered face images of 4 persons.

[14] B. Recht, "A simpler approach to matrix completion", Journal of Machine Learning Research, pp. 3413-3430, 2011.

[15] D. Gross, "Recovering low-rank matrices from few coefficients in any basis", *IEEE Trans. on Information Theory*, 57(3):1548-1566, 2011.

[16] V. Chandrasekaran, S. Sanghavi, P. Parrilo and A. Willsky, "Rank-sparsity incoherence for matrix decomposition", *SIAM Journal on Optimization*, 21 (2):572-596, 2011.

[17] Y. Hu, D. Zhang, J. Ye, X. Li and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(9): 2117-2130, 2013.

[18] Z. Lin, C. Xu, H. Zha, "Robust matrix factorization by majorization minimization", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 40(1): 208-220, 2018.

[19] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma, "TILT: Transform-invariant low-rank textures", *International Journal of Computer Vision*, 99(1): 1-24, 2012.

[20] G. Huang, V. Jain, and E. Learned-Miller, "Unsupervised joint alignment of complex images", *In Proceedings of International Conference on Computer Vision* pp. 1-8, 2007.

[21] E. Learned-Miller, "Data driven image models through continuous joint alignment", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(2):236-250, 2006.

[22] M. Cox, S. Lucey, S. Sridharan, and J. Cohn, "Least squares congealing for unsupervised alignment of images", *In Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[23] A. Vedaldi, G. Guidi, and S. Soatto, "Joint alignment up to (lossy) transforamtions", *In Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[24] Y. Fu, J. Gao, D. Tien, Z. Lin, X. Hong, "Tensor LRR and sparse coding-based subspace clustering, *IEEE Trans. on Neural Networks and Learning Systems*, 27(10): 2120-2133, 2016.

[25] W. Chen, N. Song, "Low-rank tensor completion: A Pseudo-Bayesian Learning Approach, *In Proceedings of International Conference on Computer Vision*, pp. 1-8, 2017.

[26] P. Zhou,J. Feng, "Outlier-robust tensor pca, *In Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2017.

[27] J. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics", *Linear Algebra and its Applications*, 18(2): 95-138, 1977.

[28] T. Kolda and B. Bader, "Tensor decompositions and applications", *SIAM Review*, 51(3): 455-500, 2009.

[29] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low- N-rank tensor recovery via convex optimization", *Inverse Problem*, 2011.

[30] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(1): 208-220, 2013.

[31] R. Tomioka, K. Hayashi, and H. Kashima, "Estimation of low-rank tensors via convex optimization", *Technical report*, arXiv:1010.0789, 2011.

[32] Y. Li, J. Yan, Y. Zhou, and J. Yang, "Optimum subspace learning and error correction for tensors", *In Proceedings of European Conference on Computer Vision*, pp. 790-803, 2010.

[33] K. Mohan, M. Fazel, "Iterative reweighted algorithms for matrix rank minimization", *Journal of Machine Learning Research*, 13: 3441-3473, 2012.

[34] X. Chen, F. Xu, Y. Ye, "Lower bound theory of nonzero entries in solutions of l2-lp minimization", *SIAM Journal on Scientific Computing*, 32: 2832-2852, 2010.

[35] L. Qin, Z. Lin, Y. She, C. Zhang, "A comparison of typical lp minimization algorithms", *Neurocomputing*, 119: 413-424, 2013.

[36] W. Zuo, D. Meng, L. Zhang, X. Feng, D. Zhang, "A generalized iterated shrinkage algorithm for non-convex sparse coding", *In Proceedings of International Conference on Computer Vision*, pp. 217-224, 2013.

[37] F. Nie, H. Huang, C. Ding, "Low-rank matrix recovery via efficient Schatten p-norm minimization", *In Proceedings of AAAI Conference on Artificial Intelligence*, pp. 655-661, 2012.

[38] P. Bullen, D. Mitrinovic, and P. Vasić, "Means and Their Inequalities", *East European Series*, 1988.

[39] E. Schechter, "Handbook of Analysis and its Foundations", *Academic Press*, 1996.

[40] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices", *Technical Report UILU-ENG-09-2215*, UIUC Technical Report, 2009.

[41] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers", *Foundations and Trends in Machine Learning*, 3(1): 1-122, 2011.

[42] J. Yang and X. Yuan, "Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization", *Mathematics of Computation*, 82(281): 301-329, 2013.

[43] X. Zhang, D. Wang, Z. Zhou, Y. Ma, "Simultaneous rectification and alignment via robust recovery of low-rank tensors", *In Proceedings of Advances in Neural Information Processing Systems*, pp. 1637-1645, 2013.

[44] A. Ganesh, K. Min, J. Wright and Y. Ma, "Principal component pursuit

with reduced linear measurements", *In Proceedings of International Symposium on Information Theory*, pp. 1281-1285, 2012.
[45] J. Wright, A. Ganesh, K. Min and Y. Ma, "Compressive principal component pursuit", *Information and Inference*, 2(1):32-68, 2013.
[46] Z. Lin, M. Chen and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices", *arXiv preprint arXiv:1009.5055*, 2010.
[47] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix", *Computational Advances in Multi-Sensor Adaptive Processing*, 2009.
[48] S. Ma, "Alternating proximal gradient method for convex minimization", *Preprint of Optimization Online*, 2012.
[49] Z. Luo, "On the linear convergence of the alternating direction method of multipliers", *arXiv preprint arXiv:1208.3922*, 2012.
[50] M. Signoretto, L. Lathauwer, and J. Suykens, "Nuclear norms for tensors and their use for convex multilinear estimation", *Linear Algebra and Its Applications*, 2010.
[51] F. Kittaneh, "Norm inequalities for certain operator sums", *Journal of Functional Analysis*, 1997

8 APPENDIX

8.1 Proof of Theorem 3

We prove Theorem 3 for $0 < p \leq 1$, which contains Theorem 1 as a special case.

Proof: Again, by optimality, we have

$$||\mathcal{L}^*||_p^p + \gamma||\mathcal{A}\circ\Gamma^* - \mathcal{L}^*||_{p,p}^p \leq ||\mathcal{L}_0||_p^p + \gamma||\mathcal{A}\circ\Gamma^* - \mathcal{L}_0||_{p,p}^p. \tag{31}$$

We start by recalling an important property of the Schatten-p norm of matrices [51]: for any two matrices A, B, $\|A+B\|_p^p \leq \|A\|_p^p + \|B\|_p^p$ (not that $\|A+B\|_p \leq \|A\|_p + \|B\|_p$ does not hold since Schatten-p norm is only a quasi-norm when $0 < p < 1$. Also note that both inequalities hold when $p = 1$: in this case, it is simply a nuclear norm). Consequently, Equation (31) implies that

$$\begin{aligned}||\mathcal{A}\circ\Gamma^* - \mathcal{L}^*||_{p,p}^p &\leq \frac{1}{\gamma}(||\mathcal{L}_0||_p^p - ||\mathcal{L}^*||_p^p) + ||\mathcal{A}\circ\Gamma^* - \mathcal{L}_0||_{p,p}^p \\ &\leq \frac{1}{\gamma}||\mathcal{L}_0 - \mathcal{L}^*||_p^p + ||\mathcal{A}\circ\Gamma^* - \mathcal{L}_0||_{p,p}^p,\end{aligned} \tag{32}$$

where the last inequality follows from the linearity property in the definition of $\|\cdot\|_p^p$ on tensors. This inequality then allows us to bound $||\mathcal{L}^* - \mathcal{L}_0||_{p,p}^p$ as follows:

$$\begin{aligned}||\mathcal{L}_0 - \mathcal{L}^*||_{p,p}^p &\leq ||\mathcal{A}\circ\Gamma^* - \mathcal{L}^*||_{p,p}^p + ||\mathcal{A}\circ\Gamma^* - \mathcal{L}_0||_{p,p}^p \\ &\leq \frac{1}{\gamma}(||\mathcal{L}_0 - \mathcal{L}^*||_p^p) + 2||\mathcal{A}\circ\Gamma^* - \mathcal{L}_0||_{p,p}^p \\ &= \frac{1}{\gamma}\sum_{i=1}^{N}\alpha_i||(\mathcal{L}_0 - \mathcal{L}^*)_{(i)}||_p^p + 2||\mathcal{A}\circ\Gamma^* - \mathcal{L}_0||_{p,p}^p \\ &= \frac{1}{\gamma}\sum_{i=1}^{N}\alpha_i\Big(\sum_{k=1}^{r_i}(\sigma_k^{(i)})^p\Big) + 2||\mathcal{A}\circ\Gamma^* - \mathcal{L}_0||_{p,p}^p.\end{aligned} \tag{33}$$

where the second inequality follows from substituting the inequality (32) into the current inequality, r_i is the rank of the matrix $(\mathcal{L}_0 - \mathcal{L}^*)_{(i)}$, $\sigma_1^{(i)}, \sigma_2^{(i)}, \ldots, \sigma_{r_i}^{(i)}$ are the r_i singular values of the matrix $(\mathcal{L}_0 - \mathcal{L}^*)_{(i)}$.

By Lemma 1, and setting $w_j = \frac{1}{r_i}, \forall j = 1, \ldots, r_i$, we have:

$$\begin{aligned}&\frac{(\sigma_1^{(i)})^p + (\sigma_2^{(i)})^p + \cdots + (\sigma_{r_i}^{(i)})^p}{r_i} \leq \\ &\left(\sqrt{\frac{(\sigma_1^{(i)})^2 + (\sigma_2^{(i)})^2 + \cdots + (\sigma_{r_i}^{(i)})^2}{r_i}}\right)^p,\end{aligned} \tag{34}$$

thereby leading to

$$\begin{aligned}\sum_{k=1}^{r_i}(\sigma_k^{(i)})^p &\leq r_i^{1-\frac{p}{2}}\Big(\sum_{k=1}^{r_i}(\sigma_k^{(i)})^2\Big)^{\frac{p}{2}} = r_i^{1-\frac{p}{2}}||(\mathcal{L}_0 - \mathcal{L}^*)_{(i)}||_F^p \\ &= r_i^{1-\frac{p}{2}}||\mathcal{L}_0 - \mathcal{L}^*||_F^p.\end{aligned} \tag{35}$$

Plugging this inequality into the final line in (33) results:

$$||\mathcal{L}^* - \mathcal{L}_0||_{p,p}^p \leq \frac{1}{\gamma}\sum_{i=1}^{N}\alpha_i r_i^{1-\frac{p}{2}}||\mathcal{L}_0 - \mathcal{L}^*||_F^p + 2mT^p, \tag{36}$$

since $||\mathcal{A} - \mathcal{L}_0||_{p,p}^p = ||\mathcal{E}_0||_{p,p}^p \leq mT^p$, which is itself a consequence of the generalized power-mean inequality in Lemma 1 (by setting $s = p, t = 1$):

$$\Big(\frac{||\mathcal{E}_0||_{p,p}^p}{m}\Big)^{\frac{1}{p}} \leq \frac{||\mathcal{E}_0||_{1,1}^1}{m} \leq T. \tag{37}$$

Next, we show that $||\mathcal{L}_0 - \mathcal{L}^*||_F^p \leq ||\mathcal{L}^* - \mathcal{L}_0||_{p,p}^p$. Denoting $\mathcal{L} = \mathcal{L}_0 - \mathcal{L}^*$ and using the fact that $||\mathcal{L}||_F \leq ||\mathcal{L}||_1$, we have:

$$\begin{aligned}\|\mathcal{L}\|_F &= \sqrt{\sum_{i_1,\ldots,i_N}\mathcal{L}_{i_1,\ldots,i_N}^2} \leq \sum_{i_1,\ldots,i_N}|\mathcal{L}_{i_1,\ldots,i_N}| \\ &= \Big((\sum_{i_1,\ldots,i_N}|\mathcal{L}_{i_1,\ldots,i_N}|)^p\Big)^{\frac{1}{p}} \leq \Big(\sum_{i_1,\ldots,i_N}|\mathcal{L}_{i_1,\ldots,i_N}|^p\Big)^{\frac{1}{p}} \\ &= \|\mathcal{L}\|_{p,p},\end{aligned} \tag{38}$$

where the second inequality follows from the fact that $f(x) = x^p, 0 < p < 1$ is a sub-additive function (as mentioned before). Raising both sides to the power of p yields $||\mathcal{L}_0 - \mathcal{L}^*||_F^p \leq ||\mathcal{L}^* - \mathcal{L}_0||_{p,p}^p$. Combining this inequality with Equation (36), we obtain:

$$||\mathcal{L}_0 - \mathcal{L}^*||_F^p \leq \frac{1}{\gamma}\sum_{i=1}^{N}\alpha_i r_i^{1-\frac{p}{2}}||\mathcal{L}_0 - \mathcal{L}^*||_F^p + 2mT^p \tag{39}$$

Rearranging terms, we get $||\mathcal{L}_0 - \mathcal{L}^*||_F^p \leq \frac{2mT^p}{1-\frac{1}{\gamma}\sum_{i=1}^{N}\alpha_i r_i^{1-\frac{p}{2}}} \leq \frac{2mT^p}{1-\frac{1}{\gamma}\sum_{i=1}^{N}\alpha_i I_i^{1-\frac{p}{2}}}$ (since $r_i \leq I_i$ and $1 - \frac{p}{2} > 0$), and therefore

$$||\mathcal{L}_0 - \mathcal{L}^*||_F \leq \frac{2m^{\frac{1}{p}}T}{\sqrt[p]{1 - \frac{1}{\gamma}\sum_{i=1}^{N}\alpha_i I_i^{1-\frac{p}{2}}}}, \tag{40}$$

provided that $1 > \frac{1}{\gamma}\sum_{i=1}^{N}\alpha_i I_i^{1-\frac{p}{2}}$.

□

8.2 Global Convergence of Proximal Gradient

In this part, we study the global convergence of the proximal gradient algorithm in solving the optimization problem (41) and prove Theorem 5.

$$\begin{aligned}\min_{\mathcal{L},\mathcal{E},M_i,\Delta\tilde{\Gamma}} \quad & \sum_{i=1}^{3}\alpha_i||M_i||_*+\gamma||\mathcal{E}||_1\\ \text{s.t.} \quad & \mathcal{A}\circ\Gamma+\Delta\tilde{\Gamma}=\mathcal{L}+\mathcal{E}\\ & L_{(i)}=M_i,\quad i=1,2,3\end{aligned} \tag{41}$$

We start by fixing the notation. Vectorizing matrices and tensors by taking their columns and stacking them on one another to form vectors, we denote by $\boldsymbol{m}_i$, $\boldsymbol{l}$, $\boldsymbol{e}$, $\Delta\tilde{\boldsymbol{\tau}}$, $\boldsymbol{q}_i$, $\boldsymbol{y}$ and $\boldsymbol{a}_0$ to be the associated vector representations of M_i, $\mathcal{L}$, $\mathcal{E}$, $\Delta\tilde{\boldsymbol{\Gamma}}$, Q_i, $\mathcal{Y}$ and $\mathcal{A}\circ\Gamma$ respectively. Let $\boldsymbol{x}=\begin{pmatrix}\boldsymbol{m}_1\\ \boldsymbol{m}_2\\ \boldsymbol{m}_3\\ \boldsymbol{l}\\ \boldsymbol{e}\end{pmatrix}$, $\boldsymbol{\eta}=\begin{pmatrix}\boldsymbol{y}\\ \boldsymbol{q}_1\\ \boldsymbol{q}_2\\ \boldsymbol{q}_3\end{pmatrix}$, $A=\begin{pmatrix}0&0&0&I&I\\ -I&0&0&P_1&0\\ 0&-I&0&P_2&0\\ 0&0&-I&P_3&0\end{pmatrix}$, $B=\begin{pmatrix}-I\\0\\0\\0\end{pmatrix}$, and $\boldsymbol{d}=\begin{pmatrix}\boldsymbol{a}_0\\0\\0\\0\end{pmatrix}$, where P_1, P_2 and P_3 are permutation matrices. Then the optimization problem (41) can be written as

$$\begin{aligned}\min_{\boldsymbol{x},\Delta\tilde{\boldsymbol{\tau}}} \quad & f(\boldsymbol{x})+g(\Delta\tilde{\boldsymbol{\tau}})\\ \text{s.t.} \quad & A\boldsymbol{x}+B\Delta\tilde{\boldsymbol{\tau}}=\boldsymbol{d}\ ,\end{aligned} \tag{42}$$

where $f(\boldsymbol{x})$ is vector form of $\sum_{i=1}^{3}\alpha_i||M_i||_*+\gamma||\mathcal{E}||_1$, $g(\Delta\tilde{\boldsymbol{\tau}})=0$. To prove the global convergence, we first establish the following lemma.

Lemma 2. *Assume that $(\boldsymbol{x}^*,\Delta\tilde{\boldsymbol{\tau}}^*)$ is an optimal solution of (42) and $\boldsymbol{\eta}^*$ is the corresponding optimal Lagrange multipliers. If the step sizes $\tau_1<1/\lambda_{\max}(A^\top A)$ and $\tau_2<1/\lambda_{\max}(B^\top B)$, where $\lambda_{\max}(C)$ denotes the largest eigenvalue of matrix C. Then there exists $\zeta>0$ such that the sequence $(\boldsymbol{x}^k,\Delta\tilde{\boldsymbol{\tau}}^k,\boldsymbol{\eta}^k)$ produced by Section 3.2 satisfies*

$$\|\boldsymbol{u}^k-\boldsymbol{u}^*\|_K^2-\|\boldsymbol{u}^{k+1}-\boldsymbol{u}^*\|_K^2\geq\zeta\|\boldsymbol{u}^k-\boldsymbol{u}^{k+1}\|_K^2, \tag{43}$$

where $\boldsymbol{u}^*=\begin{pmatrix}\boldsymbol{x}^*\\ \Delta\tilde{\boldsymbol{\tau}}^*\\ \boldsymbol{\eta}^*\end{pmatrix}$, $\boldsymbol{u}^k=\begin{pmatrix}\boldsymbol{x}^k\\ \Delta\tilde{\boldsymbol{\tau}}^k\\ \boldsymbol{\eta}^k\end{pmatrix}$ and $K=\begin{pmatrix}\frac{1}{\mu\tau_1}I-\frac{1}{\mu}A^\top A&0&0\\ 0&\frac{1}{\mu\tau_2}I&0\\ 0&0&\mu I\end{pmatrix}$. Note that the largest eigenvalue of matrix $(A^\top A)$ is 5, then $\tau_1<1/5$ guarantees that K is positive definite. The norm $\|\cdot\|_K^2$ is defined as $\|\boldsymbol{u}\|_K^2=\langle\boldsymbol{u},K\boldsymbol{u}\rangle$ and the corresponding inner product $\langle\boldsymbol{u},\boldsymbol{v}\rangle_K=\langle\boldsymbol{u},K\boldsymbol{v}\rangle$.

Proof: Since $(\boldsymbol{x}^*,\Delta\tilde{\boldsymbol{\tau}}^*)$ is an optimal solution of (42) and $\boldsymbol{\eta}^*$ is the corresponding optimal Lagrange multipliers, the following equations hold from the KKT conditions:

$$0\in\partial f(\boldsymbol{x}^*)-A^\top\boldsymbol{\eta}^* \tag{44}$$

$$0\in\partial g(\Delta\tilde{\boldsymbol{\tau}}^*)-B^\top\boldsymbol{\eta}^* \tag{45}$$

$$0=A\boldsymbol{x}^*+B\Delta\tilde{\boldsymbol{\tau}}^*-\boldsymbol{d} \tag{46}$$

Note that the optimality conditions for the subproblems with respect variables M_i, $\mathcal{L}$ and $\mathcal{E}$ in Section 3.2 satisfy

$$0\in\tau_1\mu\partial f(\boldsymbol{x}^{k+1})+\boldsymbol{x}^{k+1}-\boldsymbol{x}^k+\tau_1A^\top(A\boldsymbol{x}^k+B\Delta\tilde{\boldsymbol{\tau}}^k-\boldsymbol{d}-\mu\boldsymbol{\eta}^k)\ . \tag{47}$$

By using the updating formula with respect to $\mathcal{Y}$ and Q_i in Section 3.2, i.e.,

$$\boldsymbol{\eta}^{k+1}=\boldsymbol{\eta}^k-(A\boldsymbol{x}^{k+1}+B\Delta\tilde{\boldsymbol{\tau}}^{k+1}-\boldsymbol{d})/\mu\ , \tag{48}$$

(47) can be reduced to

$$\begin{aligned}0\in&\tau_1\mu\partial f(\boldsymbol{x}^{k+1})+\boldsymbol{x}^{k+1}-\boldsymbol{x}^k+\\ &\tau_1A^\top(A\boldsymbol{x}^k-A\boldsymbol{x}^{k+1}+B\Delta\tilde{\boldsymbol{\tau}}^k-B\Delta\tilde{\boldsymbol{\tau}}^{k+1}-\mu\boldsymbol{\eta}^{k+1})\ .\end{aligned} \tag{49}$$

Combining (44) and (49) and using the fact that $\partial f(\cdot)$ is a monotone operator, we obtain

$$\begin{aligned}(\boldsymbol{x}^{k+1}-\boldsymbol{x}^*)^\top\Big(&\frac{1}{\tau_1\mu}(\boldsymbol{x}^k-\boldsymbol{x}^{k+1})-\frac{1}{\mu}A^\top A(\boldsymbol{x}^k-\boldsymbol{x}^{k+1})\\ &-\frac{1}{\mu}A^\top B(\Delta\tilde{\boldsymbol{\tau}}^k-\Delta\tilde{\boldsymbol{\tau}}^{k+1})+A^\top(\boldsymbol{\eta}^{k+1}-\boldsymbol{\eta}^*)\Big)\geq 0\ .\end{aligned} \tag{50}$$

The optimality conditions for the subproblem with respect variable $\Delta\tilde{\Gamma}$ in Section 3.2 satisfy

$$\begin{aligned}0\in&\tau_2\mu\partial g(\Delta\tilde{\boldsymbol{\tau}}^{k+1})+\Delta\tilde{\boldsymbol{\tau}}^{k+1}-\Delta\tilde{\boldsymbol{\tau}}^k+\\ &\tau_2B^\top(A\boldsymbol{x}^{k+1}+B\Delta\tilde{\boldsymbol{\tau}}^k-\boldsymbol{d}-\mu\boldsymbol{\eta}^k)\ .\end{aligned} \tag{51}$$

Using (48), (51) can be reduced to

$$\begin{aligned}0\in&\tau_2\mu\partial g(\Delta\tilde{\boldsymbol{\tau}}^{k+1})+\Delta\tilde{\boldsymbol{\tau}}^{k+1}-\Delta\tilde{\boldsymbol{\tau}}^k+\\ &\tau_2B^\top(B\Delta\tilde{\boldsymbol{\tau}}^k-B\Delta\tilde{\boldsymbol{\tau}}^{k+1}-\mu\boldsymbol{\eta}^{k+1})\ .\end{aligned} \tag{52}$$

Combining (45) and (52) and using the fact that $\partial g(\cdot)$ is a monotone operator, we obtain

$$\begin{aligned}(\Delta\tilde{\boldsymbol{\tau}}^{k+1}-\Delta\tilde{\boldsymbol{\tau}}^*)^\top\Big(&\frac{1}{\tau_2\mu}(\Delta\tilde{\boldsymbol{\tau}}^k-\Delta\tilde{\boldsymbol{\tau}}^{k+1})\\ &-\frac{1}{\mu}B^\top B(\Delta\tilde{\boldsymbol{\tau}}^k-\Delta\tilde{\boldsymbol{\tau}}^{k+1})+B^\top(\boldsymbol{\eta}^{k+1}-\boldsymbol{\eta}^*)\Big)\geq 0\ .\end{aligned} \tag{53}$$

Summing (50) and (53), and using $A\boldsymbol{x}^*+B\Delta\tilde{\boldsymbol{\tau}}^*=\boldsymbol{d}$, we get

$$\begin{aligned}&\frac{1}{\tau_1\mu}(\boldsymbol{x}^{k+1}-\boldsymbol{x}^*)^\top(\boldsymbol{x}^k-\boldsymbol{x}^{k+1})-\frac{1}{\mu}(\boldsymbol{x}^{k+1}-\boldsymbol{x}^*)^\top A^\top A(\boldsymbol{x}^k-\boldsymbol{x}^{k+1})\\ &+\frac{1}{\tau_2\mu}(\Delta\tilde{\boldsymbol{\tau}}^{k+1}-\Delta\tilde{\boldsymbol{\tau}}^*)^\top(\Delta\tilde{\boldsymbol{\tau}}^k-\Delta\tilde{\boldsymbol{\tau}}^{k+1})\\ &-(\boldsymbol{\eta}^k-\boldsymbol{\eta}^{k+1})^\top B(\Delta\tilde{\boldsymbol{\tau}}^k-\Delta\tilde{\boldsymbol{\tau}}^{k+1})\\ &+\mu(\boldsymbol{\eta}^k-\boldsymbol{\eta}^{k+1})^\top(\boldsymbol{\eta}^{k+1}-\boldsymbol{\eta}^*)\geq 0.\end{aligned} \tag{54}$$

Using the notation of $\boldsymbol{u}^k$, $\boldsymbol{u}^*$ and K, (54) can be written as

$$\langle\boldsymbol{u}^{k+1}-\boldsymbol{u}^*,\boldsymbol{u}^k-\boldsymbol{u}^{k+1}\rangle_K\geq\langle\boldsymbol{\eta}^k-\boldsymbol{\eta}^{k+1},B\Delta\tilde{\boldsymbol{\tau}}^k-B\Delta\tilde{\boldsymbol{\tau}}^{k+1}\rangle\ . \tag{55}$$

which can be further written as

$$\begin{aligned}&\langle\boldsymbol{u}^k-\boldsymbol{u}^*,\boldsymbol{u}^k-\boldsymbol{u}^{k+1}\rangle_K\\ \geq\ &\|\boldsymbol{u}^k-\boldsymbol{u}^{k+1}\|_K^2+\langle\boldsymbol{\eta}^k-\boldsymbol{\eta}^{k+1},B\Delta\tilde{\boldsymbol{\tau}}^k-B\Delta\tilde{\boldsymbol{\tau}}^{k+1}\rangle\ .\end{aligned} \tag{56}$$

Combining (56) with the identity

$$\begin{aligned}\|\boldsymbol{u}^{k+1}-\boldsymbol{u}^*\|_K^2 &= \|\boldsymbol{u}^{k+1}-\boldsymbol{u}^k\|_K^2+\|\boldsymbol{u}^k-\boldsymbol{u}^*\|_K^2\\ &\quad -2\langle \boldsymbol{u}^k-\boldsymbol{u}^*,\boldsymbol{u}^k-\boldsymbol{u}^{k+1}\rangle_K\ , \end{aligned}\tag{57}$$

we get

$$\begin{aligned}&\|\boldsymbol{u}^k-\boldsymbol{u}^*\|_K^2-\|\boldsymbol{u}^{k+1}-\boldsymbol{u}^*\|_K^2\\ =\ & 2\langle \boldsymbol{u}^k-\boldsymbol{u}^*,\boldsymbol{u}^k-\boldsymbol{u}^{k+1}\rangle_K-\|\boldsymbol{u}^{k+1}-\boldsymbol{u}^k\|_K^2\\ \le\ & \|\boldsymbol{u}^{k+1}-\boldsymbol{u}^k\|_K^2+2\langle \boldsymbol{\eta}^k-\boldsymbol{\eta}^{k+1},B\Delta\tilde{\boldsymbol{\tau}}^k-B\Delta\tilde{\boldsymbol{\tau}}^{k+1}\rangle. \end{aligned}\tag{58}$$

Let $\xi=\frac{1+\tau_2}{2}$, then we know that $\tau_2<\xi<1$ since $\tau_2<1$. Let $\rho=\mu\xi$, Cauchy-Schwartz inequality then implies:

$$\begin{aligned}&2\langle \boldsymbol{\eta}^k-\boldsymbol{\eta}^{k+1},B\Delta\tilde{\boldsymbol{\tau}}^k-B\Delta\tilde{\boldsymbol{\tau}}^{k+1}\rangle\\ \ge\ & -\rho\|\boldsymbol{\eta}^k-\boldsymbol{\eta}^{k+1}\|^2-\frac{1}{\rho}\|B\Delta\tilde{\boldsymbol{\tau}}^k-B\Delta\tilde{\boldsymbol{\tau}}^{k+1}\|^2\\ =\ & -\rho\|\boldsymbol{\eta}^k-\boldsymbol{\eta}^{k+1}\|^2-\frac{1}{\rho}\|\Delta\tilde{\boldsymbol{\tau}}^k-\Delta\tilde{\boldsymbol{\tau}}^{k+1}\|^2\ . \end{aligned}\tag{59}$$

Combining (58) and (59) we get

$$\begin{aligned}&\|\boldsymbol{u}^k-\boldsymbol{u}^*\|_K^2-\|\boldsymbol{u}^{k+1}-\boldsymbol{u}^*\|_K^2\\ \ge\ & (\boldsymbol{x}^k-\boldsymbol{x}^{k+1})^\top\left(\frac{1}{\tau_1\mu}I-\frac{1}{\mu}A^\top A\right)(\boldsymbol{x}^k-\boldsymbol{x}^{k+1})\\ &+\left(\frac{1}{\mu\tau_2}-\frac{1}{\rho}\right)\|\Delta\tilde{\boldsymbol{\tau}}^k-\Delta\tilde{\boldsymbol{\tau}}^{k+1})\|^2+(\mu-\rho)\|\boldsymbol{\eta}^k-\boldsymbol{\eta}^{k+1}\|^2\\ \ge\ & \zeta\|\boldsymbol{u}^k-\boldsymbol{u}^{k+1}\|_K^2 \end{aligned}\tag{60}$$

where $\zeta=\min\{\frac{1}{\tau_1\mu}-\frac{2}{\mu},\frac{1}{\mu\tau_2}-\frac{1}{\rho},\mu-\rho\}>0$ since the largest eigenvalue of matrix $(A^\top A)$ is 5 and $\rho<\mu$. The proof is complete. □

Finally, we prove the global convergence result of the proximal gradient algorithm.

Theorem 6. *The sequence $\{(\boldsymbol{x}^k,\Delta\tilde{\boldsymbol{\tau}}^k,\boldsymbol{\eta}^k)\}$ generated by the proximal gradient descent scheme in Section 3.2 with $\tau_1<\frac{1}{5}$ and $\tau_2<1$ converges to the optimal solution to problem (41).*

Proof: From Lemma 2 we can easily get that

(i) $\|\boldsymbol{u}^k-\boldsymbol{u}^{k+1}\|_K\to 0$;
(ii) $\{\boldsymbol{u}^k\}$ lies in a compact region;
(iii) $\|\boldsymbol{u}^k-\boldsymbol{u}^*\|_K$ is monotonically non-increasing and thus converges.

It follows from (i) that $\boldsymbol{x}^k-\boldsymbol{x}^{k+1}\to 0$, $\Delta\tilde{\boldsymbol{\tau}}^k-\Delta\tilde{\boldsymbol{\tau}}^{k+1}\to 0$ and $\boldsymbol{\eta}^k-\boldsymbol{\eta}^{k+1}\to 0$. Then (48) implies that $A\boldsymbol{x}^k+B\Delta\tilde{\boldsymbol{\tau}}-\boldsymbol{d}\to 0$. From (ii) we obtain that, $\{\boldsymbol{u}^k\}$ has a subsequence $\{\boldsymbol{u}^{k_j}\}$ that converges to $\boldsymbol{u}_0=(\boldsymbol{x}_0,\Delta\tilde{\boldsymbol{\tau}}_0,\boldsymbol{\eta}_0)$. Therefore, $(\boldsymbol{x}_0,\Delta\tilde{\boldsymbol{\tau}}_0,\boldsymbol{\eta}_0)$ is a limit point of $\{(\boldsymbol{x}^k,\Delta\tilde{\boldsymbol{\tau}}^k,\boldsymbol{\eta}^k)\}$ and $A\boldsymbol{x}_0+B\Delta\tilde{\boldsymbol{\tau}}_0=\boldsymbol{d}$.

Note that (49) implies that

$$0\in\partial f(\boldsymbol{x}_0)-A^\top\boldsymbol{\eta}_0\ .\tag{61}$$

Note also that (52) implies that

$$0\in\partial g(\Delta\tilde{\boldsymbol{\tau}}_0)-B^\top\boldsymbol{\eta}_0\ .\tag{62}$$

(61) and (62) and $A\boldsymbol{x}_0+B\Delta\tilde{\boldsymbol{\tau}}_0=\boldsymbol{d}$ imply that $(\boldsymbol{x}_0,\Delta\tilde{\boldsymbol{\tau}}_0,\boldsymbol{\eta}_0)$ satisfies the KKT conditions for (42).

To complete the proof, it remains to show that $\{(\boldsymbol{x}^k,\Delta\tilde{\boldsymbol{\tau}}^k,\boldsymbol{\eta}^k)\}$ has a unique limit point. Let $(\boldsymbol{x}_0,\Delta\tilde{\boldsymbol{\tau}}_0,\boldsymbol{\eta}_0)$ and $(\boldsymbol{x}_1,\Delta\tilde{\boldsymbol{\tau}}_1,\boldsymbol{\eta}_1)$ be any two limit points of sequence $\{(\boldsymbol{x}^k,\Delta\tilde{\boldsymbol{\tau}}^k,\boldsymbol{\eta}^k)\}$. As we have shown, both $(\boldsymbol{x}_0,\Delta\tilde{\boldsymbol{\tau}}_0,\boldsymbol{\eta}_0)$ and $(\boldsymbol{x}_1,\Delta\tilde{\boldsymbol{\tau}}_1,\boldsymbol{\eta}_1)$ are optimal solutions to (42). Thus, $\boldsymbol{u}^*$ in (43) can be replaced by $\boldsymbol{u}_0$ and $\boldsymbol{u}_1$. This leads to

$$\|\boldsymbol{u}^{k+1}-\boldsymbol{u}_i\|_K^2\le\|\boldsymbol{u}^k-\boldsymbol{u}_i\|_K^2,\qquad i=0,1,$$

and we thus get the existence of the limits

$$\lim_{k\to\infty}\|\boldsymbol{u}^k-\boldsymbol{u}_i\|_K^2=\nu_i<+\infty,\qquad i=0,1,$$

Now using the identity

$$\|\boldsymbol{u}^k-\boldsymbol{u}_0\|_K^2-\|\boldsymbol{u}^k-\boldsymbol{u}_1\|_K^2=-2\langle\boldsymbol{u}^k,\boldsymbol{u}_0-\boldsymbol{u}_1\rangle_K+\|\boldsymbol{u}_0\|_K^2-\|\boldsymbol{u}_1\|_K^2$$

and passing the limit we get

$$\nu_0^2-\nu_1^2=-2\langle\boldsymbol{u}_0,\boldsymbol{u}_0-\boldsymbol{u}_1\rangle_K+\|\boldsymbol{u}_0\|_K^2-\|\boldsymbol{u}_1\|_K^2=-\|\boldsymbol{u}_0-\boldsymbol{u}_1\|_K^2;$$

$$\nu_0^2-\nu_1^2=-2\langle\boldsymbol{u}_1,\boldsymbol{u}_0-\boldsymbol{u}_1\rangle_K+\|\boldsymbol{u}_0\|_K^2-\|\boldsymbol{u}_1\|_K^2=\|\boldsymbol{u}_0-\boldsymbol{u}_1\|_K^2.$$

Thus we must have $\|\boldsymbol{u}_0-\boldsymbol{u}_1\|_K^2=0$ and hence the limit point of $\{(\boldsymbol{x}^k,\Delta\tilde{\boldsymbol{\tau}}^k,\boldsymbol{\eta}^k)\}$ is unique. □