

Color image recovery using low-rank quaternion matrix completion algorithm

Jifei Miao* and Kit Ian Kou†

Department of Mathematics, Faculty of Science and Technology, University of Macau, Macau 999078, China

Abstract

As a new color image representation tool, quaternion has achieved excellent results in color image processing problems. In this paper, we propose a novel low-rank quaternion matrix completion algorithm to recover missing data of color image. Motivated by two kinds of low-rank approximation approaches (low-rank decomposition and nuclear norm minimization) in traditional matrix-based methods, we combine the two approaches in our quaternion matrix-based model. Furthermore, the nuclear norm of the quaternion matrix is replaced by the sum of Frobenius norm of its two low-rank factor quaternion matrices. Based on the relationship between quaternion matrix and its equivalent complex matrix, the problem eventually is converted from quaternion number field to complex number field. An alternating minimization method is applied to solve the model. Simulation results on real world color image recovery show the superior performance and efficiency of the proposed algorithm over some state-of-the-art tensor-based ones.

Keywords: Color image recovery, quaternion, matrix completion, low-rank decomposition.

1 Introduction

Color images are applied in numerous fields, from the casual documentation of events to medical applications. A color image contains red, blue, and green channels. In most cases, some data of the acquired images are missed during acquisition and transmission. Hence, a well-performed recovery technology should be proposed.

In the past few decades, low-rank matrix completion problem has been widely studied and proven very useful in the application of image recovery [1–5]. Commonly, the method is to stack all the image pixels as column vectors of a matrix, and recovery theory and algorithm are adopted to the resulting matrix which is low-rank or approximately low-rank. However, these image recovery models are usually developed for gray-level images. For color image processing,

*jifmiao@163.com

†kikou@umac.mo

traditional matrix-based methods usually ignore the mutual connection among channels, because these recovery methods are applied to red, green, and blue channels separately, which is likely to result in color distortion during the recovery process.

More recently, an increasing number of low-rank tensor completion methods have been proposed to recover color images [6–11]. Actually, a tensor is a natural form of high-dimensional and multi-way real world data. For example, a color image can be regarded as a 3-way tensor due to its three channels, each frontal slice of this 3-way tensor corresponds to a channel of the color image. The state-of-the-art methods for tensor completion mainly consist of approaches of two types. The first type is nuclear norm minimization [8, 12, 13], which is generally computationally expensive and time-consuming, for example some of this type of algorithms require computing tensor singular value decomposition (t-SVD) which costs much computation especially for natural large scale data. The second type involves the use of low-rank tensor decomposition techniques, for example some Tucker decomposition based techniques have been proposed in [8, 14], and some CP (CANDECOMP/PARAFAC) decomposition based techniques have been proposed in [6, 7]. Nevertheless, for this type of algorithms, the rank of a tensor is generally pretty hard to determine [10], so they usually cannot offer the best low-rank approximation to a tensor. In brief, the recovery theory for low-rank tensor completion problem is not well established compared with that of matrix-based completion problem.

Different from conventional matrix-based and tensor-based models, in this paper, we propose a novel low-rank quaternion matrix completion algorithm to recover missing data of color image. Actually, the Red, Green and Blue values of each pixel of a color image can be naturally represented as a single pure quaternion valued pixel [15].

$$q(x, y) = r(x, y)i + g(x, y)j + b(x, y)k, \quad (1)$$

where $r(x, y)$, $g(x, y)$ and $b(x, y)$ are, respectively, the red, green and blue components corresponding to the pixel at position (x, y) in the color image, and i , j and k are the three imaginary units. By using (1), an $M \times N$ color image is described by a matrix with size $M \times N$ whose elements are pure quaternions. The main advantage of this representation is that it processes a colour image holistically as a vector field and handles the coupling between the color channels naturally [16–18], and color information of source image is fully used. Moreover, comparing to the tensor-based model, the quaternion-based model not only preserves the correlation among channels but also the orthogonal property for the coefficients of different channels, which achieves a structured representation [19]. Hence, as a new color image representation tool, quaternion has achieved excellent results in the color image processing including color image filtering [16], color image edge detection [20], color image denoising [21], color image watermarking [22], color face recognition [23], color image recovery [24] and so on.

However, the quaternion matrix completion based color image recovery problem has been less investigated. In [24], the authors proposed a quaternion matrix completion algorithm by solving a semi-definite programming optimization (SDP) problem which can be solved by the interior-point method. In many practical applications, nevertheless, the matrices are very large, which makes the SDP problem intractable [2]. When the size of the matrices exceeds 100×100 , the existing state-of-the-art SDP solvers such as SDPT3 [25] and SeDuMi [26] are generally no longer applicable. In this paper, based on low-rank decomposition of quaternion matrix, we propose a novel low-rank matrix completion algorithm in quaternion number field. To the best of our knowledge, the quaternion matrix completion problem based on low-rank decomposition

has not yet been investigated. In summary, our main contributions include:

- We propose an efficient low-rank quaternion matrix completion algorithm to recover missing data of color image. Both low-rank decomposition and nuclear norm minimization techniques are combined in our quaternion matrix-based model. The nuclear norm of the quaternion matrix is replaced by the sum of Frobenius norm of its two low-rank factor quaternion matrices. Furthermore, based on the relationship between the quaternion matrix and its equivalent complex matrix, the problem eventually be converted from quaternion number field to complex number field. An alternating minimization method is applied to solve the model, which is easily implemented and has low computational complexity.
- We adopt the rank-decreasing method to estimate the rank of a matrix. Convergence and complexity of the proposed algorithm is analyzed. Experimental results demonstrate the effectiveness of the proposed algorithm in color image recovery. Moreover, we compare the proposed approach with several state-of-the-art tensor-based ones. The results validate the competitive performance of the proposed algorithm compared with the other approaches.

The remainder of this paper is organized as follows. Section 2 introduces some notations and preliminaries for quaternion algebra. Section 3 reviews the matrix completion theory and proposes our quaternion-based matrix completion model. The detailed overview of quaternion matrix completion algorithm is presented in Section IV. Section V provides some experiments to illustrate the performance of our algorithm, and compare it with some state-of-the-art methods. Finally, some conclusions are drawn in Section VI.

2 Notations and preliminaries

In this section, we first summarize some main notations and then introduce some basic knowledge of quaternion algebra.

2.1 Notations

In this paper, \mathbb{R} , \mathbb{C} and \mathbb{H} respectively denote the set of real numbers, the set of complex numbers and the set of quaternions. A scalar, a vector, a matrix, and a tensor are written as a , \mathbf{a} , \mathbf{A} , and \mathcal{A} , respectively. For a tensor \mathcal{A} , we use the Matlab notation $\mathcal{A}(:, :, k)$ to denote its k -th frontal slice. \tilde{a} , $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{A}}$ respectively represent a quaternion scalar, a quaternion vector and a quaternion matrix. $(\cdot)^*$, $(\cdot)^{-1}$, $(\cdot)^\dagger$, $(\cdot)^T$ and $(\cdot)^H$ denote the conjugation, inverse, Moore-Penrose inverse, transpose and conjugate transpose respectively. $|\cdot|$, $\|\cdot\|_1$, $\|\cdot\|_F$ and $\|\cdot\|_*$ are respectively the absolute value or modulus, the l_1 norm, the Frobenius norm and the nuclear norm¹. \circ denotes the inner product operation. $\text{tr}\{\cdot\}$ and $\text{rank}(\cdot)$ denote the trace and rank operators respectively. \mathbf{I}_K represents the identity matrix of size $K \times K$. And we denote $\text{diag}(\mathbf{x})$ as a diagonal matrix whose diagonal elements are as same as those of \mathbf{x} .

¹For (quaternion) matrix, the nuclear norm is defined as the sum of its singular values. For tensor, the nuclear norm is defined as the nuclear norms of mode matrices.

2.2 Basic knowledge of quaternion algebra

As a natural extension of the complex space, the quaternion space was first introduced by W. Hamilton [27] in 1843. A quaternion $\tilde{q} \in \mathbb{H}$ is composed of a real component and three imaginary components.

$$\tilde{q} = q_0 + q_1i + q_2j + q_3k, \quad (2)$$

where $q_l \in \mathbb{R}$ ($l = 0, 1, 2, 3$) are real coefficients, i, j, k are imaginary number units and obey the quaternion rules that

$$\begin{cases} i^2 = j^2 = k^2 = ijk = -1, \\ ij = -ji = k, \\ jk = -kj = i, \\ ki = -ik = j. \end{cases} \quad (3)$$

If the real component $q_0 = 0$, \tilde{q} is named a pure quaternion. Every quaternion $\tilde{q} = q_0 + q_1i + q_2j + q_3k$ can be uniquely represented as $\tilde{q} = q_0 + q_1i + (q_2 + q_3i)j = c_1 + c_2j$, where c_1 and c_2 are complex numbers.

The conjugate and the modulus of a quaternion \tilde{q} are, respectively, defined as follows:

$$\tilde{q}^* = q_0 - q_1i - q_2j - q_3k, \quad (4)$$

$$|\tilde{q}| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}. \quad (5)$$

Unlike complex number systems, the product of two quaternions \tilde{q}_1 and \tilde{q}_2 is noncommutative, *i.e.*, $\tilde{q}_1\tilde{q}_2 \neq \tilde{q}_2\tilde{q}_1$ in general.

Analogously, a quaternion matrix $\ddot{\mathbf{Q}} = (\ddot{q}_{mn}) \in \mathbb{H}^{M \times N}$ is written as $\ddot{\mathbf{Q}} = \mathbf{Q}_0 + \mathbf{Q}_1i + \mathbf{Q}_2j + \mathbf{Q}_3k$, where $\mathbf{Q}_l \in \mathbb{R}^{M \times N}$ ($l = 0, 1, 2, 3$), $\ddot{\mathbf{Q}}$ is named a pure quaternion matrix when $\mathbf{Q}_0 = \mathbf{0}$. The Frobenius norm of the quaternion matrix is defined as $\|\ddot{\mathbf{Q}}\|_F = \sqrt{\sum_{m=1}^M \sum_{n=1}^N |\ddot{q}_{mn}|^2} = \sqrt{\text{tr}\{(\ddot{\mathbf{Q}})^H \ddot{\mathbf{Q}}\}}$.

The most common way to study quaternion matrices is to use their complex representation. Given a quaternion matrix $\ddot{\mathbf{Q}} \in \mathbb{H}^{M \times N}$, it can be uniquely expressed as $\ddot{\mathbf{Q}} = \mathbf{Q}_a + \mathbf{Q}_bj$, where $\mathbf{Q}_a, \mathbf{Q}_b \in \mathbb{C}^{M \times N}$. We define the operator $f : \mathbb{H}^{M \times N} \longrightarrow \mathbb{C}^{2M \times 2N}$, then the complex representation matrix of $\ddot{\mathbf{Q}} = \mathbf{Q}_a + \mathbf{Q}_bj \in \mathbb{H}^{M \times N}$ is denoted as follows [28]:

$$f(\ddot{\mathbf{Q}}) = \begin{pmatrix} \mathbf{Q}_a & \mathbf{Q}_b \\ -\mathbf{Q}_b^* & \mathbf{Q}_a^* \end{pmatrix}, \quad (6)$$

$f(\ddot{\mathbf{Q}})$ is uniquely determined by $\ddot{\mathbf{Q}}$. Denote f^{-1} as the inverse operator of f . Note that if \mathbf{Q} is a complex matrix or real matrix, then

$$f(\mathbf{Q}) = \begin{pmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^* \end{pmatrix} \quad (7)$$

or

$$f(\mathbf{Q}) = \begin{pmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{pmatrix}, \quad (8)$$

respectively.

There are some properties of f (see Theorem 1 and Theorem 2).

Theorem 1. [29] Let $\ddot{\mathbf{P}} \in \mathbb{H}^{M \times N}$, $\ddot{\mathbf{Q}} \in \mathbb{H}^{M \times N}$, then

1. $f(\ddot{\mathbf{P}}\ddot{\mathbf{Q}}) = f(\ddot{\mathbf{P}})f(\ddot{\mathbf{Q}})$,
2. $f(\ddot{\mathbf{P}} + \ddot{\mathbf{Q}}) = f(\ddot{\mathbf{P}}) + f(\ddot{\mathbf{Q}})$,
3. $f(\ddot{\mathbf{P}}^*) = f(\ddot{\mathbf{P}})^*$,
4. $f(\ddot{\mathbf{P}}^{-1}) = f(\ddot{\mathbf{P}})^{-1}$, if \mathbf{P}^{-1} exists,
5. $\|f(\ddot{\mathbf{P}})\|_F^2 = 2\|\ddot{\mathbf{P}}\|_F^2$,
6. $\ddot{\mathbf{P}}$ is unitary, Hermitian, or normal if and only if $f(\ddot{\mathbf{P}})$ is unitary, Hermitian, or normal, respectively.

Theorem 2. Let $\ddot{\mathbf{P}} \in \mathbb{H}^{M \times N}$, we have $\text{rank}(\ddot{\mathbf{P}}) = \frac{1}{2}\text{rank}(f(\ddot{\mathbf{P}}))$.

The proof of Theorem 2 can be found in Appendix A. Readers can find more details on quaternion algebra in [29–31].

3 Problem formulation

In this section, we first review the matrix completion theory and then propose our quaternion-based matrix completion model.

3.1 Matrix completion theory

Matrix completion problem consists of recovering a matrix from a subset of its entries. The usual structural assumption on a matrix that makes the problem well posed is that the matrix is low-rank or approximate low-rank. The optimization model for matrix completion was proposed firstly in [32], and can be formulated as:

$$\begin{aligned} & \underset{\mathbf{X}}{\text{minimize}} \quad \text{rank}(\mathbf{X}) \\ & \text{subject to} \quad \mathcal{P}_\Omega(\mathbf{X} - \mathbf{T}) = \mathbf{0}, \end{aligned} \tag{9}$$

where \mathbf{X} is a completed output matrix, \mathbf{T} is an incomplete input matrix and the Ω is the entries set, more concretely, if \mathbf{X}_{mn} is observed, then $(m, n) \in \Omega$, and \mathcal{P}_Ω is the unitary projection onto the linear space of matrices supported on Ω , defined as

$$(\mathcal{P}_\Omega(\mathbf{X}))_{mn} = \begin{cases} \mathbf{X}_{mn}, & (m, n) \in \Omega, \\ 0, & (m, n) \notin \Omega. \end{cases}$$

Because such rank minimization problem (9) is generally NP-hard [33], various heuristics algorithms have been developed to solve this problem. These methods could be divided into two main categories: nuclear norm minimization method, see *e.g.* [34–36] and low-rank matrix decomposition approach, see *e.g.* [37, 38].

The matrix nuclear norm is a popular convex surrogate of non-convex rank function, and its minimization method is widely used in practice:

$$\begin{aligned} & \underset{\mathbf{X}}{\text{minimize}} && \|\mathbf{X}\|_* \\ & \text{subject to} && \mathcal{P}_\Omega(\mathbf{X} - \mathbf{T}) = \mathbf{0}. \end{aligned} \quad (10)$$

The nuclear norm minimization problem (10) is generally solved iteratively in which singular value decompositions (SVD) is involved at each iteration. So the nuclear norm minimization methods bear the computational cost required by SVD which becomes increasingly expensive as the sizes of the matrices increase [10, 38]. Hence, a non-SVD approach, *i.e.* low-rank matrix decomposition, has been proposed in order to more efficiently solve large-scale matrix completion problems.

The low-rank matrix decomposition-based completion problem is formulated in the form of the following optimization problem [37]:

$$\begin{aligned} & \underset{\mathbf{U}, \mathbf{V}, \mathbf{X}}{\text{minimize}} && \frac{1}{2} \|\mathbf{UV} - \mathbf{X}\|_F^2 \\ & \text{subject to} && \mathcal{P}_\Omega(\mathbf{X} - \mathbf{T}) = \mathbf{0}, \end{aligned} \quad (11)$$

where $\mathbf{U} \in \mathbb{C}^{M \times K}$, $\mathbf{V} \in \mathbb{C}^{K \times N}$, $\mathbf{X} \in \mathbb{C}^{M \times N}$, and the integer K is the rank of matrix \mathbf{X} .

3.2 Proposed formulation of quaternion matrix completion

Quaternion matrix completion can be regard as the generalization of the traditional matrix completion in the quaternion number field, which is to fill in the missing values of a quaternion matrix $\ddot{\mathbf{X}} \in \mathbb{H}^{M \times N}$ under a given subset Ω of its entries $\{\ddot{\mathbf{X}}_{m,n} | (m, n) \in \Omega\}$. Motivated by traditional matrix completion techniques, low-rank decomposition and nuclear norm minimization, we combine the two approaches and propose our quaternion matrix completion model. Before that, we first present the following theorem:

Theorem 3. *Suppose that $\ddot{\mathbf{X}} \in \mathbb{H}^{M \times N}$, $\ddot{\mathbf{P}} \in \mathbb{H}^{M \times N}$ and $\ddot{\mathbf{Q}} \in \mathbb{H}^{N \times M}$ are three arbitrary quaternion matrices. Then, we have the following properties:*

- (1) *If $\text{rank}(\ddot{\mathbf{X}}) = K$, then there exists two quaternion matrices $\ddot{\mathbf{U}} \in \mathbb{H}^{M \times K}$ and $\ddot{\mathbf{V}} \in \mathbb{H}^{K \times N}$ such that*

$$\ddot{\mathbf{X}} = \ddot{\mathbf{U}}\ddot{\mathbf{V}},$$

and they satisfy

$$\text{rank}(\ddot{\mathbf{U}}) = \text{rank}(\ddot{\mathbf{V}}) = K;$$

- (2) $\text{rank}(\ddot{\mathbf{P}}\ddot{\mathbf{Q}}) \leq \min(\text{rank}(\ddot{\mathbf{P}}), \text{rank}(\ddot{\mathbf{Q}}));$

- (3) *Assume $\ddot{\mathbf{X}} = \ddot{\mathbf{U}}\ddot{\mathbf{V}}$ is a completed output quaternion matrix, $\ddot{\mathbf{T}}$ is an incomplete input quaternion matrix with $\text{rank } K_0 \leq K$. The nuclear norm minimization problem*

$$\begin{aligned} & \underset{\ddot{\mathbf{X}}}{\text{minimize}} && \|f(\ddot{\mathbf{X}})\|_* \\ & \text{subject to} && \mathcal{P}_\Omega(\ddot{\mathbf{X}} - \ddot{\mathbf{T}}) = \mathbf{0} \end{aligned} \quad (12)$$

is equivalent to the following quadratic optimization problem:

$$\begin{aligned} & \underset{f(\ddot{\mathbf{U}}), f(\ddot{\mathbf{V}})}{\text{minimize}} && \frac{1}{2} \left(\|f(\ddot{\mathbf{U}})\|_F^2 + \|f(\ddot{\mathbf{V}})\|_F^2 \right) \\ & \text{subject to} && \mathcal{P}_\Omega(\ddot{\mathbf{X}} - \ddot{\mathbf{T}}) = \mathbf{0}. \end{aligned} \quad (13)$$

The proof of Theorem 3 can be found in Appendix B. Thus, based on the properties (1) and (2) in Theorem 3, similar to the matrix decomposition method, we can adopt a low-rank quaternion matrix decomposition strategy to deal with the large scale quaternion matrix completion problem more efficiently. Furthermore, we also consider the nuclear norm $\|\ddot{\mathbf{X}}\|_*$ in our model but replaced by $\|\ddot{\mathbf{U}}\|_F^2 + \|\ddot{\mathbf{V}}\|_F^2$ according to property (3) in Theorem 3.

Accordingly, the low-rank quaternion matrix completion formulation can be written as follows:

$$\begin{aligned} & \underset{\ddot{\mathbf{U}}, \ddot{\mathbf{V}}, \ddot{\mathbf{X}}}{\text{minimize}} && \frac{1}{2} \|\ddot{\mathbf{U}}\ddot{\mathbf{V}} - \ddot{\mathbf{X}}\|_F^2 + \frac{\lambda}{2} \left(\|\ddot{\mathbf{U}}\|_F^2 + \|\ddot{\mathbf{V}}\|_F^2 \right) \\ & \text{subject to} && \mathcal{P}_\Omega(\ddot{\mathbf{X}} - \ddot{\mathbf{T}}) = \mathbf{0}, \end{aligned} \quad (14)$$

where λ is a nonnegative parameter.

4 Proposed algorithm

In this section, we first show how to solve the optimization problem (14), then we introduce a rank-decreasing method to adjust the rank of a matrix. Finally, we provide the convergence and complexity analyses of the proposed algorithm.

4.1 Optimization process

On account of the noncommutativity of the multiplication in quaternion space, the definition and computation of the gradient of quaternion matrix function are generally much more complicated than those in complex space [39], which hugely increases the difficulty to handle the quaternion-based optimization problems. Therefore, based on the defined operator f in (6) and its properties in Theorem 1, we tend to convert the problem (14) to that in the complex number field and reformulate as follows, which is differentiable and separable among its blocks:

$$\begin{aligned} & \underset{f(\ddot{\mathbf{U}}), f(\ddot{\mathbf{V}}), \ddot{\mathbf{X}}}{\text{minimize}} && \frac{1}{2} \|f(\ddot{\mathbf{U}})f(\ddot{\mathbf{V}}) - f(\ddot{\mathbf{X}})\|_F^2 \\ & && + \frac{\lambda}{2} \left(\|f(\ddot{\mathbf{U}})\|_F^2 + \|f(\ddot{\mathbf{V}})\|_F^2 \right) \\ & \text{subject to} && \mathcal{P}_\Omega(\ddot{\mathbf{X}} - \ddot{\mathbf{T}}) = \mathbf{0}. \end{aligned} \quad (15)$$

Note that $f(\ddot{\mathbf{U}}) \in \mathbb{C}^{2M \times 2K}$ and $f(\ddot{\mathbf{V}}) \in \mathbb{C}^{2K \times 2N}$ are all complex-valued matrices.

Although, it is obvious that problem (15) is non-convex itself, it is convex with respect to each single variable. Hence, we adopt a simple but efficient iterative scheme to solve the optimization problem (15) by using an alternating minimization approach. More specifically, we update only one of the variables $f(\ddot{\mathbf{U}})$, $f(\ddot{\mathbf{V}})$ and $\ddot{\mathbf{X}}$ each time while remaining the other two fixed, and three variables all will be updated sequentially and iteratively.

Letting

$$\begin{aligned}\mathcal{G}(f(\ddot{\mathbf{U}}), f(\ddot{\mathbf{V}}), \ddot{\mathbf{X}}) &= \frac{1}{2} \|f(\ddot{\mathbf{U}})f(\ddot{\mathbf{V}}) - f(\ddot{\mathbf{X}})\|_F^2 \\ &\quad + \frac{\lambda}{2} \left(\|f(\ddot{\mathbf{U}})\|_F^2 + \|f(\ddot{\mathbf{V}})\|_F^2 \right),\end{aligned}\tag{16}$$

we perform the updates as

$$f(\ddot{\mathbf{U}})^{\tau+1} = \arg \min_{f(\ddot{\mathbf{U}})} \mathcal{G}(f(\ddot{\mathbf{U}}), f(\ddot{\mathbf{V}})^\tau, \ddot{\mathbf{X}}^\tau),\tag{17a}$$

$$f(\ddot{\mathbf{V}})^{\tau+1} = \arg \min_{f(\ddot{\mathbf{V}})} \mathcal{G}(f(\ddot{\mathbf{U}})^{\tau+1}, f(\ddot{\mathbf{V}}), \ddot{\mathbf{X}}^\tau),\tag{17b}$$

$$\ddot{\mathbf{X}}^{\tau+1} = \arg \min_{\mathcal{P}_\Omega(\ddot{\mathbf{X}} - \ddot{\mathbf{T}}) = \mathbf{0}} \mathcal{G}(f(\ddot{\mathbf{U}})^{\tau+1}, f(\ddot{\mathbf{V}})^{\tau+1}, \ddot{\mathbf{X}}),\tag{17c}$$

where τ is the iteration index.

By introducing a Lagrange multiplier Υ for the constraint $\mathcal{P}_\Omega(\ddot{\mathbf{X}} - \ddot{\mathbf{T}}) = \mathbf{0}$, the Lagrangian function of (15) is defined as

$$\begin{aligned}\mathcal{Q}(f(\ddot{\mathbf{U}}), f(\ddot{\mathbf{V}}), \ddot{\mathbf{X}}, \Upsilon) &= \mathcal{G}(f(\ddot{\mathbf{U}}), f(\ddot{\mathbf{V}}), \ddot{\mathbf{X}}) \\ &\quad - \Upsilon \circ \mathcal{P}_\Omega(\ddot{\mathbf{X}} - \ddot{\mathbf{T}}).\end{aligned}$$

Differentiating the function $\mathcal{Q}(f(\ddot{\mathbf{U}}), f(\ddot{\mathbf{V}}), \ddot{\mathbf{X}}, \Upsilon)$, we have the following Karush-Kuhn-Tucker (KKT) Conditions:

$$(f(\ddot{\mathbf{U}})f(\ddot{\mathbf{V}}) - f(\ddot{\mathbf{X}}))f(\ddot{\mathbf{V}})^H + \lambda f(\ddot{\mathbf{U}}) = \mathbf{0},\tag{18a}$$

$$f(\ddot{\mathbf{U}})^H(f(\ddot{\mathbf{U}})f(\ddot{\mathbf{V}}) - f(\ddot{\mathbf{X}})) + \lambda f(\ddot{\mathbf{V}}) = \mathbf{0},\tag{18b}$$

$$\mathcal{P}_{\Omega^c} \left(\ddot{\mathbf{X}} - f^{-1}(f(\ddot{\mathbf{U}})f(\ddot{\mathbf{V}})) \right) = \mathbf{0},\tag{18c}$$

$$\mathcal{P}_\Omega(\ddot{\mathbf{X}} - \ddot{\mathbf{T}}) = \mathbf{0},\tag{18d}$$

$$\mathcal{P}_\Omega \left(\ddot{\mathbf{X}} - f^{-1}(f(\ddot{\mathbf{U}})f(\ddot{\mathbf{V}})) \right) - \Upsilon = \mathbf{0}.\tag{18e}$$

Thus, the updates in (17) can be explicitly written as follows:

$$f(\ddot{\mathbf{U}})^{\tau+1} = f(\ddot{\mathbf{X}})^\tau (f(\ddot{\mathbf{V}})^\tau)^H \Psi_{\ddot{\mathbf{V}}},\tag{19}$$

$$f(\ddot{\mathbf{V}})^{\tau+1} = \Phi_{\ddot{\mathbf{U}}} (f(\ddot{\mathbf{U}})^{\tau+1})^H f(\ddot{\mathbf{X}})^\tau,\tag{20}$$

where

$$\Psi_{\ddot{\mathbf{V}}} = \left(f(\ddot{\mathbf{V}})^\tau (f(\ddot{\mathbf{V}})^\tau)^H + \lambda \mathbf{I}_{2K} \right)^\dagger,$$

$$\Phi_{\ddot{\mathbf{U}}} = \left((f(\ddot{\mathbf{U}})^{\tau+1})^H f(\ddot{\mathbf{U}})^{\tau+1} + \lambda \mathbf{I}_{2K} \right)^\dagger.$$

Then, we can directly obtain $\ddot{\mathbf{X}}^{\tau+1}$ as

$$\ddot{\mathbf{X}}^{\tau+1} = \mathcal{P}_{\Omega^c} \left(f^{-1}(f(\ddot{\mathbf{U}})^{\tau+1})f(\ddot{\mathbf{V}}^{\tau+1}) \right) + \ddot{\mathbf{T}},\tag{21}$$

where Ω^c is the complement of Ω , and we have used the fact that $\mathcal{P}_{\Omega^c}(\ddot{\mathbf{T}}) = \mathbf{0}$ in (21).

4.2 Rank estimation based on a rank-decreasing method

A proper estimation to the rank $2K$ (labeled as r) for the model (15) is essential for the success of the proposed algorithm. Although the target rank can be adjusted manually, it would be usually time-consuming for large-scale data. We introduce a rank-decreasing method to estimate the rank, which is similar to that in [38].

This method starts from an input overestimated rank r of $f(\ddot{\mathbf{X}})$, *i.e.*, $r > \text{rank}(f(\ddot{\mathbf{X}}))$. Suppose that the rank of $f(\ddot{\mathbf{X}})^\tau$ is r^τ . We compute the eigenvalues of $(f(\ddot{\mathbf{U}})^\tau)^H f(\ddot{\mathbf{U}})^\tau \mathbf{\Pi}^\tau$, where $\mathbf{\Pi}^\tau$ is a permutation matrix so that all these eigenvalues are ordered non-increasing, *i.e.*, $\mathbf{d}_1^\tau \geq \mathbf{d}_2^\tau \geq \dots \geq \mathbf{d}_{r^\tau}^\tau$. Then, we compute the quotient sequence $\hat{\mathbf{d}}_m^\tau = \mathbf{d}_m^\tau / \mathbf{d}_{m+1}^\tau$, ($m = 1, \dots, r^\tau - 1$). Assume that

$$p^\tau = \arg \max_{1 \leq m \leq r^\tau - 1} \hat{\mathbf{d}}_m^\tau,$$

and define

$$\mu^\tau = \frac{(r^\tau - 1) \hat{\mathbf{d}}_{p^\tau}^\tau}{\sum_{m \neq p^\tau} \hat{\mathbf{d}}_m^\tau}. \quad (22)$$

If $\mu^\tau \geq 10$, *i.e.*, there being a large drop in the estimated rank of the $f(\ddot{\mathbf{X}})^\tau$, we should reduce r^τ to p^τ . Then, assuming $\mathbf{L}^\tau \mathbf{\Sigma}^\tau (\mathbf{R}^H)^\tau$ is the SVD of $f(\ddot{\mathbf{U}})^\tau f(\ddot{\mathbf{V}})^\tau$, we can update $f(\ddot{\mathbf{U}})^\tau = \mathbf{L}_{p^\tau}^\tau \mathbf{\Sigma}_{p^\tau}^\tau$ and $f(\ddot{\mathbf{V}})^\tau = (\mathbf{R}^H)_{p^\tau}^\tau$, where $\mathbf{L}_{p^\tau}^\tau$ consists of the first p^τ columns of $f(\ddot{\mathbf{U}})^\tau$, and $(\mathbf{R}^H)_{p^\tau}^\tau$ and $\mathbf{\Sigma}_{p^\tau}^\tau$ are obtained accordingly. Note that, doing only one time of this rank-adjusting scheme is generally enough during the whole iterative process. Hence, the computational complexity generated from the SVD of $f(\ddot{\mathbf{U}})^\tau f(\ddot{\mathbf{V}})^\tau$ in this rank-adjusting process is negligible relative to that of the whole iterative process.

Finally, the proposed **Low-Rank Quaternion Matrix Completion (LRQMC)** algorithm can be summarized as shown in TABLE 1.

Table 1: The low-rank quaternion matrix completion (LRQMC) algorithm.

Input: The quaternion matrix data $\ddot{\mathbf{X}} \in \mathbb{H}^{M \times N}$, the observed set Ω , and the initialized rank r^0 .
1: Initialize $f(\ddot{\mathbf{U}})^0 \in \mathbb{C}^{2M \times r^0}$ and $f(\ddot{\mathbf{V}})^0 \in \mathbb{C}^{r^0 \times 2N}$ randomly, and appropriate parameter $\lambda > 0$.
2: Repeat
3: Fix $f(\ddot{\mathbf{V}})^\tau$ and $f(\ddot{\mathbf{X}})^\tau$ to update $f(\ddot{\mathbf{U}})^{\tau+1}$ by (19), <i>i.e.</i> , $f(\ddot{\mathbf{U}})^{\tau+1} \leftarrow f(\ddot{\mathbf{X}})^\tau (f(\ddot{\mathbf{V}})^\tau)^H \Psi_{\ddot{\mathbf{V}}}$.
4: Fix $f(\ddot{\mathbf{U}})^{\tau+1}$ and $f(\ddot{\mathbf{X}})^\tau$ to update $f(\ddot{\mathbf{V}})^{\tau+1}$ by (20), <i>i.e.</i> , $f(\ddot{\mathbf{V}})^{\tau+1} \leftarrow \Phi_{\ddot{\mathbf{U}}}(f(\ddot{\mathbf{U}})^{\tau+1})^H f(\ddot{\mathbf{X}})^\tau$.
5: Fix $f(\ddot{\mathbf{U}})^{\tau+1}$ and $f(\ddot{\mathbf{V}})^{\tau+1}$ to update $\ddot{\mathbf{X}}^{\tau+1}$ by (21), <i>i.e.</i> , $\ddot{\mathbf{X}}^{\tau+1} \leftarrow \mathcal{P}_{\Omega^c} \left(f^{-1}(f(\ddot{\mathbf{U}})^{\tau+1}) f(\ddot{\mathbf{V}})^{\tau+1} \right) + \ddot{\mathbf{T}}$.
6: if $\mu^\tau \geq 10$ in (22) then
7: Apply rank-decreasing method to adjust r^τ and the sizes of $f(\ddot{\mathbf{U}})^{\tau+1}$ and $f(\ddot{\mathbf{V}})^{\tau+1}$.
8: end if
9: $\tau \leftarrow \tau + 1$.
10: Until convergence
Output: $f(\ddot{\mathbf{U}})^{\tau+1}$, $f(\ddot{\mathbf{V}})^{\tau+1}$ and $\ddot{\mathbf{X}}^{\tau+1}$

4.3 The convergence and the computational complexity analyses

Convergence analysis: From (18e), we can clearly see that $\mathcal{P}_\Omega(\ddot{\mathbf{X}} - f^{-1}(f(\ddot{\mathbf{U}})f(\ddot{\mathbf{V}}))) = \Upsilon$, *i.e.*, the multiplier matrix Υ measures the residual $\ddot{\mathbf{X}} - f^{-1}(f(\ddot{\mathbf{U}})f(\ddot{\mathbf{V}}))$ in Ω and thus has no effect in the process of determining $f(\ddot{\mathbf{U}})$, $f(\ddot{\mathbf{V}})$ and $\ddot{\mathbf{X}}$. Therefore, for simplicity, we just discuss $\mathcal{G}(f(\ddot{\mathbf{U}}), f(\ddot{\mathbf{V}}), \ddot{\mathbf{X}})$ in (16).

Since the Hessian matrices of $\mathcal{G}(f(\ddot{\mathbf{U}}), f(\ddot{\mathbf{V}}), \ddot{\mathbf{X}})$ *w.r.t.* $f(\ddot{\mathbf{U}})$ and $f(\ddot{\mathbf{V}})$ are respectively $f(\ddot{\mathbf{V}})f(\ddot{\mathbf{V}})^H + \lambda\mathbf{I}_{2K}$ and $f(\ddot{\mathbf{U}})f(\ddot{\mathbf{U}})^H + \lambda\mathbf{I}_{2K}$ which are positive semidefinite matrices (they are even positive definite when $\lambda > 0$). Hence, for any $\tau \geq 0$, we have $\mathcal{G}(f(\ddot{\mathbf{U}})^\tau, f(\ddot{\mathbf{V}})^\tau, \ddot{\mathbf{X}}^\tau) - \mathcal{G}(f(\ddot{\mathbf{U}})^{\tau+1}, f(\ddot{\mathbf{V}})^{\tau+1}, \ddot{\mathbf{X}}^\tau) \geq 0$. On the other hand, we note that $\ddot{\mathbf{X}}^{\tau+1}$ is the optimal solution to problem (14):

$$\begin{aligned} \ddot{\mathbf{X}}^{\tau+1} = & \arg \min_{\mathcal{P}_\Omega(\ddot{\mathbf{X}} - \ddot{\mathbf{T}}) = \mathbf{0}} \frac{1}{2} \|\ddot{\mathbf{U}}^{\tau+1} \ddot{\mathbf{V}}^{\tau+1} - \ddot{\mathbf{X}}\|_F^2 \\ & + \frac{\lambda}{2} (\|\ddot{\mathbf{U}}^{\tau+1}\|_F^2 + \|\ddot{\mathbf{V}}^{\tau+1}\|_F^2). \end{aligned} \quad (23)$$

At the same time, we note that $\mathcal{P}_\Omega(\ddot{\mathbf{X}}^{\tau+1} - \ddot{\mathbf{T}}) = \mathbf{0}$, *i.e.*, $\ddot{\mathbf{X}}^{\tau+1}$ is a feasible solution to problem (23). So the inequality $\frac{1}{2} \|\ddot{\mathbf{U}}^{\tau+1} \ddot{\mathbf{V}}^{\tau+1} - \ddot{\mathbf{X}}^{\tau+1}\|_F^2 \leq \frac{1}{2} \|\ddot{\mathbf{U}}^{\tau+1} \ddot{\mathbf{V}}^{\tau+1} - \ddot{\mathbf{X}}^\tau\|_F^2$ holds, *i.e.*, the inequality $\frac{1}{2} \|f(\ddot{\mathbf{U}})^{\tau+1} f(\ddot{\mathbf{V}})^{\tau+1} - f(\ddot{\mathbf{X}})^{\tau+1}\|_F^2 \leq \frac{1}{2} \|f(\ddot{\mathbf{U}})^{\tau+1} f(\ddot{\mathbf{V}})^{\tau+1} - f(\ddot{\mathbf{X}})^\tau\|_F^2$ holds. Hence, for any $\tau \geq 0$, we can obtain $\mathcal{G}(f(\ddot{\mathbf{U}})^{\tau+1}, f(\ddot{\mathbf{V}})^{\tau+1}, \ddot{\mathbf{X}}^\tau) - \mathcal{G}(f(\ddot{\mathbf{U}})^{\tau+1}, f(\ddot{\mathbf{V}})^{\tau+1}, \ddot{\mathbf{X}}^{\tau+1}) \geq 0$. Then, it follows that

$$\begin{aligned} & \mathcal{G}(f(\ddot{\mathbf{U}})^\tau, f(\ddot{\mathbf{V}})^\tau, \ddot{\mathbf{X}}^\tau) - \mathcal{G}(f(\ddot{\mathbf{U}})^{\tau+1}, f(\ddot{\mathbf{V}})^{\tau+1}, \ddot{\mathbf{X}}^{\tau+1}) \\ & = \mathcal{G}(f(\ddot{\mathbf{U}})^\tau, f(\ddot{\mathbf{V}})^\tau, \ddot{\mathbf{X}}^\tau) - \mathcal{G}(f(\ddot{\mathbf{U}})^{\tau+1}, f(\ddot{\mathbf{V}})^{\tau+1}, \ddot{\mathbf{X}}^\tau) \\ & \quad + \mathcal{G}(f(\ddot{\mathbf{U}})^{\tau+1}, f(\ddot{\mathbf{V}})^{\tau+1}, \ddot{\mathbf{X}}^\tau) \\ & \quad - \mathcal{G}(f(\ddot{\mathbf{U}})^{\tau+1}, f(\ddot{\mathbf{V}})^{\tau+1}, \ddot{\mathbf{X}}^{\tau+1}) \\ & \geq 0. \end{aligned}$$

Consequently, the function $\mathcal{G}(f(\ddot{\mathbf{U}})^\tau, f(\ddot{\mathbf{V}})^\tau, \ddot{\mathbf{X}}^\tau)$ decreases monotonically, and it is obvious that $\mathcal{G}(f(\ddot{\mathbf{U}})^\tau, f(\ddot{\mathbf{V}})^\tau, \ddot{\mathbf{X}}^\tau) \geq 0$, so the theoretical convergence of the proposed algorithm (LRQMC) can be guaranteed.

Computational complexity: We analyze the computational complexity within one iteration for LRQMC algorithm provided in TABLE 1. When updating $f(\ddot{\mathbf{U}})$ and $f(\ddot{\mathbf{V}})$ respectively by (19) and (20), the computational cost is about $\mathcal{O}(\hat{r}(\hat{r}^2 + N\hat{r} + M\hat{r} + MN))$, where \hat{r} is the estimated rank of $f(\ddot{\mathbf{X}})$. Then the computational cost of updating $\ddot{\mathbf{X}}$ by (21) is about $\mathcal{O}(MN\hat{r})$. In rank estimation process, the estimated rank is detected based on the economy-size QR decomposition whose computational cost is about $\mathcal{O}(\hat{r}^3)$. Hence, the total computational cost of LRQMC algorithm at each iteration is about $\mathcal{O}(\hat{r}(\hat{r}^2 + N\hat{r} + M\hat{r} + MN))$.

5 Simulation results

In this section, simulations on some natural color images are conducted to evaluate the performance of the proposed LRQMC algorithm. And we compare it with several existing state-of-the-art tensor-based methods, including TCTF [10], SPC [6], TMac (including TMac-inc and TMac-dec) [40], STDC [14] and LRTC (including FaLRTC and SiLRTC) [8]. All the simulations

are run in MATLAB 2014b under Windows 10 on a personal computer with 2.20GHz CPU and 8GB memory.

A color image is a 3-way tensor defined by two indices for spatial variables and one index for color mode [9]. All the images, in our simulation, are initially represented by 3-way tensors $\mathcal{T} \in \mathbb{R}^{M \times N \times s}$, where $M \times N$ is the number of pixels in the image and $s = 3$ is the number of colors (red, green and blue). For LRQMC algorithm, each image is reshaped as a pure quaternion matrix $\ddot{\mathbf{T}} \in \mathbb{H}^{M \times N}$ by using the following way:

$$\ddot{\mathbf{T}} = \mathcal{T}(:, :, 1)i + \mathcal{T}(:, :, 2)j + \mathcal{T}(:, :, 3)k.$$

In addition, we uniformly generate the index set Ω at Gaussian random distribution, and define the sampling ratio (SR) as:

$$\text{SR} = \frac{\text{numel}(\Omega)}{M \times N \times s},$$

where $\text{numel}(\Omega)$ represents the number of observation elements in the index set Ω .

Quantitative assessment: In order to evaluate the performance of proposed algorithm, except visual quality, we employ four quantitative quality indexes, including the relative square error (RSE), the peak signal-to-noise ratio (PSNR), the structure similarity (SSIM) and the feature similarity (FSIM), which are respectively defined as follows:

$$\text{RSE} = 10\log_{10} \left(\frac{\|\mathcal{X} - \mathcal{T}\|_F}{\|\mathcal{T}\|_F} \right),$$

where \mathcal{X} and \mathcal{T} are the recovered and truth data, respectively.

$$\text{PSNR} = 10\log_{10} \left(\frac{\text{Peakval}^2}{\text{MSE}} \right),$$

where Peakval is taken from the range of the image datatype (*e.g.*, for uint8 image it is 255), MSE is the mean square error, *i.e.* $\text{MSE} = \|\mathcal{X} - \mathcal{T}\|_F^2 / \text{numel}(\mathcal{X})$.

$$\text{SSIM} = \frac{(2\mu_{\mathcal{T}}\mu_{\mathcal{X}} + C_1)(2\sigma_{\mathcal{T}\mathcal{X}} + C_2)}{(\mu_{\mathcal{T}}^2 + \mu_{\mathcal{X}}^2 + C_1)(\sigma_{\mathcal{T}}^2 + \sigma_{\mathcal{X}}^2 + C_2)},$$

where $\mu_{\mathcal{T}}$, $\mu_{\mathcal{X}}$, $\sigma_{\mathcal{T}}$, $\sigma_{\mathcal{X}}$ and $\sigma_{\mathcal{T}\mathcal{X}}$ are the local means, standard deviations, and cross-covariance for images \mathcal{T} and \mathcal{X} , $C_1 = (0.01L)^2$, $C_2 = (0.03L)^2$, $C_3 = C_2/2$, L is the specified dynamic range of the pixel values.

$$\text{FSIM} = \frac{\sum_{z \in \Delta} S_L(z) PC_m(z)}{\sum_{z \in \Delta} PC_m(z)},$$

where Δ demotes the whole image spatial domain. The phase congruency for position z of image \mathcal{T} is denoted as $PC_x(\mathcal{T})$, then $PC_m(z) = \max\{PC_{\mathcal{T}(z)}, PC_{\mathcal{X}(z)}\}$, $S_L(z)$ is the gradient magnitude for position z .

For LRQMC algorithm, $\mathcal{X}(:, :, 1) = \text{Imag}_1(\ddot{\mathbf{T}})$, $\mathcal{X}(:, :, 2) = \text{Imag}_2(\ddot{\mathbf{T}})$ and $\mathcal{X}(:, :, 3) = \text{Imag}_3(\ddot{\mathbf{T}})$, where $\text{Imag}_n(\ddot{\mathbf{T}})$ ($n = 1, 2, 3$) denotes n -th image part of $\ddot{\mathbf{T}}$.

Datasets: In the simulations, we use two color image datasets: Berkeley segmentation dataset and Kodak PhotoCD dataset. The statistics of these two datasets are briefly summarized below:

- Berkeley Segmentation Dataset (BSD):² There are 300 clean color images of size $481 \times 321 \times 3$ in the whole dataset.
- Kodak PhotoCD Dataset (Kodak):³ The whole dataset consists of 24 clean color images of size $512 \times 768 \times 3$.

We first show that these color images can be well approximated by the low-rank quaternion matrices. Actually, as mentioned in [10, 41], when the image data is arranged into matrices or tensors, they lie on a union of low-rank subspaces approximately, which indicate the low-rank structure of the image data. This is also true for quaternion matrices data. For instance, in Fig. 1 we display the singular values of four images (reshaped as pure quaternion matrices) selected from the two color image datasets randomly. One can obviously see that most of the singular values are very close to 0, and much smaller than the first several larger singular values. So we could say that these color images can be well approximated by the low-rank quaternion matrices as we desired.

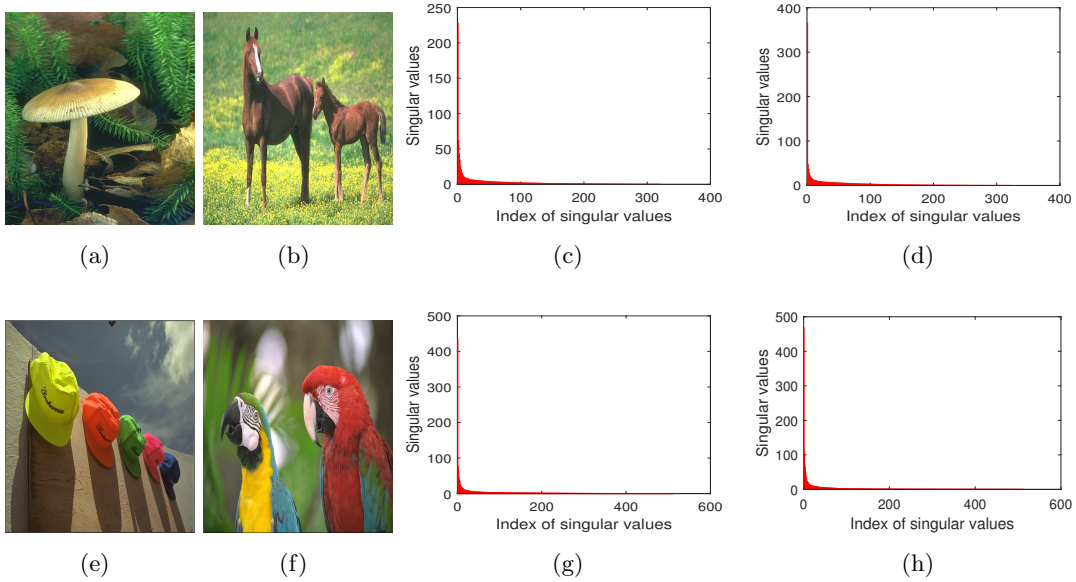


Figure 1: Illustration of the low-rank property of the images in the two color image datasets. (a) and (b) are two images randomly selected from the BSD, (c) and (d) respectively display the singular values of (a) and (b). (e) and (f) are two images randomly selected from the Kodak, (g) and (h) respectively display the singular values of (e) and (f).

Parameter settings: For our LRQMC algorithm, the initial rank r of $f(\ddot{\mathbf{X}})$ is set as $r = 50$, and we set $\lambda = 0.5$. For TCTF, we set the initialized rank $r^0 = [30, 30, 30]$ the same as that in [10]. For SPC, we use QV constraint with $\rho = [1.0, 1.0, 0]$. For TMac-inc, we set the initialized rank $r^0 = [3, 3, 3]$ with increment 2. For TMac-dec, we set the initialized rank $r^0 = [30, 30, 30]$. And we set $\alpha_n = \frac{1}{3}$, $n = 1, 2, 3$ for both TMac-inc and TMac-dec as suggested in [40]. For SiLRTC, according to [8], the weight parameter $\alpha = \theta / \|\theta\|_1$, where $\theta = [1, 1, 1e^{-3}]$. In addition, the stopping criteria for all the algorithms that we adopted are the difference between the values

²<https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>

³<http://r0k.us/graphics/kodak/>

of $\varepsilon := \|\mathcal{X} - \mathcal{T}\|_F$ in two consecutive iterations, *i.e.* $|\varepsilon^\tau - \varepsilon^{\tau+1}| < 1e^{-3}$, where τ is the iteration index, and the maximum number of iterations is 1,000.

Simulation 1: In this simulation, we use BSD dataset to evaluate our algorithm for color image recovery. We randomly select 56 color images from this dataset. 6 examples of selected images are shown in Fig. 2 (a) (from top to bottom, we label them orderly as Image (1), Image (2), Image (3), Image (4), Image (5) and Image (6)) Fig. 2 (b) is the observed image with $SR = 0.3$. Fig. 2 (c)-(h) are the recover results of LRQMC, TCTF, SPC, TMac-inc, TMac-dec and SiLRTC. We see from the Fig. 2 that the color images recovered by LRQMC are visually better than those recovered by the other compared algorithms. TABLE 2 summaries the RSE, PSNR, SSIM, FSIM values and the running time of all the algorithms on the six testing images displayed in Fig. 2 (a). From the results, one can observe that the overall performance of LRQMC is much better than that of TCTF, SPC TMac-inc and SiLRTC, and also has an advantage over TMac-dec on most images, except for the Image (5). However, the running time of TMac-dec is much longer than that of LRQMC.

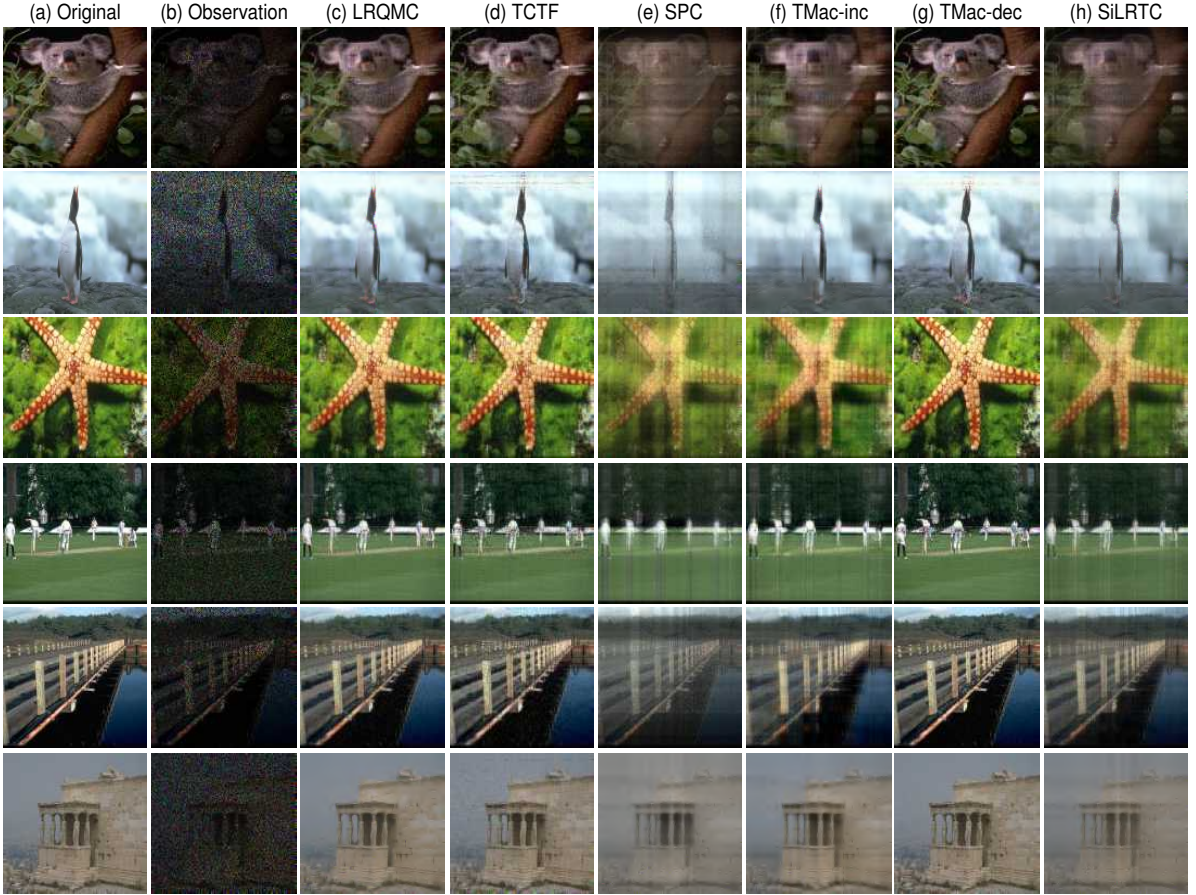


Figure 2: Examples of color image recovery using different algorithms ($SR = 0.3$).

In Fig. 3, we report the RSE, PSNR, SSIM and FSIM results of different algorithms on the remaining 50 images. From the results, one can obviously find that our LRQMC algorithm perform better than all the other algorithms in the vast majority of images.

Simulation 2: In this simulation, we use Kodak dataset to evaluate our algorithm for color

Table 2: Quantitative quality indexes and running time (seconds) of different algorithms on the six images displayed in Fig. 2 (a) (SR = 0.3).

Images	<div>Algorithms</div> <div>Indexes</div>	LRQMC	TCTF	SPC	TMac-inc	TMac-dec	SiLRTC
Image (1)	RSE	-8.519	-6.049	-5.031	-5.535	-8.039	-5.113
	PSNR	29.046	23.408	19.805	21.893	28.102	21.049
	SSIM	0.872	0.671	0.538	0.631	0.814	0.617
	FSIM	0.995	0.979	0.957	0.962	0.991	0.968
	time(s)	9.578	9.000	14.431	10.571	50.693	10.592
Image (2)	RSE	-12.858	-9.790	-8.578	-11.168	-10.868	-10.049
	PSNR	28.673	24.153	20.166	25.293	24.693	23.055
	SSIM	0.875	0.761	0.678	0.808	0.844	0.755
	FSIM	0.991	0.965	0.908	0.953	0.985	0.934
	time(s)	7.421	16.507	3.456	8.900	61.362	13.642
Image (3)	RSE	-9.213	-7.108	-5.100	-6.107	-9.138	-6.186
	PSNR	25.484	21.474	17.177	21.190	25.251	19.348
	SSIM	0.893	0.830	0.718	0.792	0.887	0.809
	FSIM	0.991	0.980	0.946	0.956	0.991	0.965
	time(s)	16.419	11.859	62.164	8.931	65.614	12.765
Image (4)	RSE	-7.257	-4.753	-5.396	-6.489	-5.489	-5.989
	PSNR	23.350	19.504	19.661	21.813	19.814	20.814
	SSIM	0.791	0.678	0.669	0.741	0.786	0.724
	FSIM	0.988	0.973	0.963	0.967	0.982	0.971
	time(s)	11.863	9.391	15.194	8.168	64.076	11.832
Image (5)	RSE	-8.097	-6.409	-5.141	-6.143	-8.639	-6.009
	PSNR	24.480	21.136	18.177	20.176	25.088	19.909
	SSIM	0.706	0.589	0.438	0.564	0.768	0.580
	FSIM	0.989	0.979	0.963	0.967	0.989	0.974
	time(s)	9.753	7.048	7.715	5.780	58.974	12.126
Image (6)	RSE	-12.315	-9.261	-8.476	-10.481	-12.268	-9.105
	PSNR	29.613	23.425	21.354	25.364	28.941	22.612
	SSIM	0.901	0.724	0.505	0.818	0.895	0.650
	FSIM	0.989	0.973	0.952	0.965	0.986	0.958
	time(s)	6.874	7.645	13.713	5.702	47.972	15.129

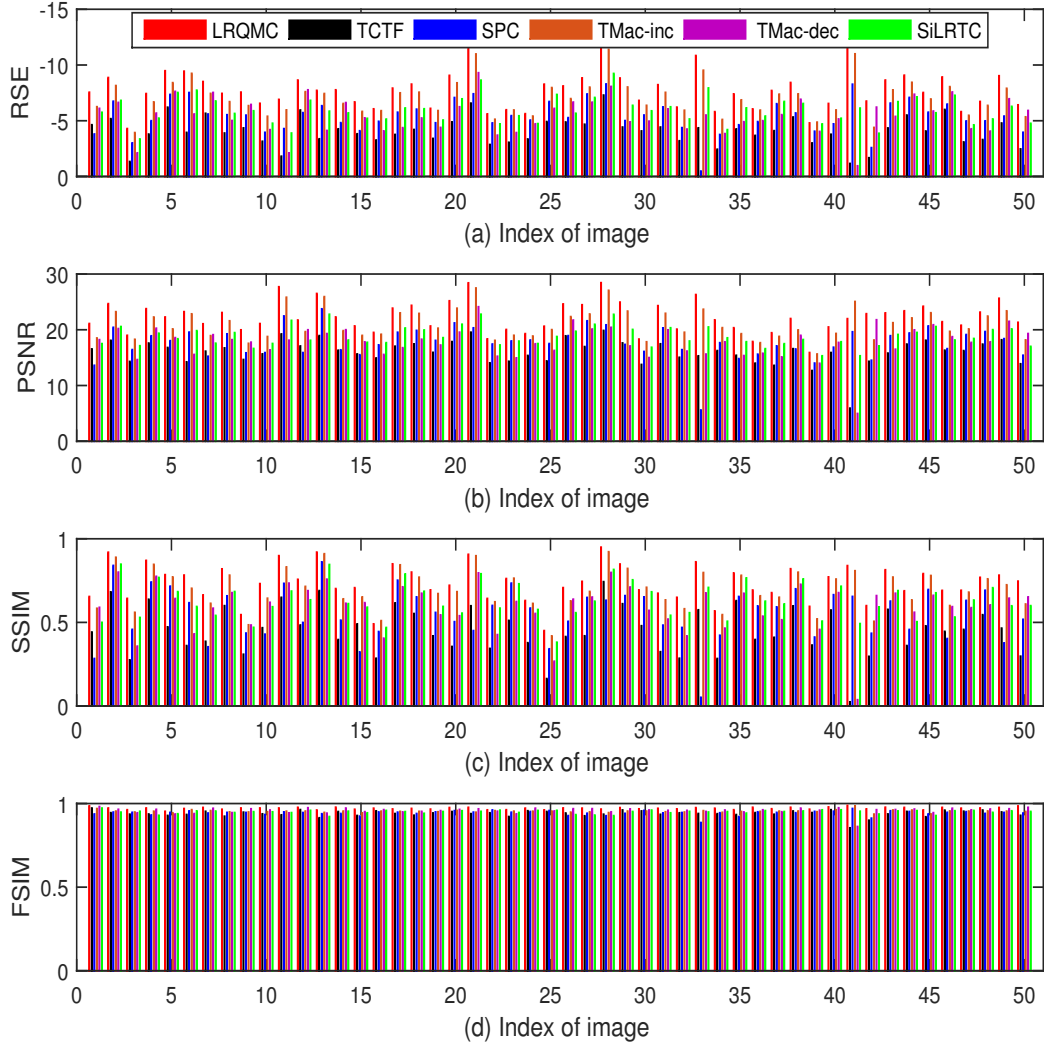


Figure 3: Comparison of RSE, PSNR, SSIM and FSIM results of different algorithms for color image recovery on 50 BSD images (SR = 0.2). The figure is viewed better in zoomed PDF.

image recovery. We randomly select 2 color images from Kodak dataset as shown in Fig. 4. Fig. 5 and Fig. 5 respectively show the recovery results of Fig. 4 (a) and Fig. 4 (b) SRs from 0.1 to 0.5 using different algorithms in terms of the RSE, PSNR, SSIM and FSIM. LRQMC outperforms all the other algorithms. Furthermore, we can also find that the merit of LRQMC is more obvious in low sampling ratio, *e.g.*, SR = 0.1.



Figure 4: Randomly selected 2 color images from Kodak dataset.

6 Conclusion

We propose a novel low-rank quaternion matrix completion algorithm to recover missing data of color image. Quaternion representation processes a colour image holistically as a vector field and handles the coupling between the color channels naturally, and color information of source image is fully used. We combine low-rank decomposition and nuclear norm (which is replaced by Frobenius norm of the two low-rank factor quaternion matrices) minimization approaches in our quaternion matrix-based model. Based on the relationship between the quaternion matrix and its equivalent complex matrix, the problem eventually be converted from quaternion number field to complex number field. An alternating minimization method is applied to solve the model, which guarantees convergence of the proposed algorithm. Simulation results on real world color image recovery demonstrate the competitive performance of the proposed algorithm compared to several state-of-the-art tensor-based methods.

A Proof of the Theorem 2

Lemma 1. (The SVD of quaternion matrix (QSVD) [29]) Let $\ddot{\mathbf{P}} \in \mathbb{H}^{M \times N}$ be of rank K . Then there exist unitary quaternion matrices⁴ $\ddot{\mathbf{A}} \in \mathbb{H}^{M \times M}$ and $\ddot{\mathbf{B}} \in \mathbb{H}^{N \times N}$ such that

$$\ddot{\mathbf{A}}\ddot{\mathbf{P}}\ddot{\mathbf{B}} = \begin{pmatrix} \mathbf{\Sigma}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

where $\mathbf{\Sigma}_K = \text{diag}(\sigma_1, \dots, \sigma_K)$ is a real diagonal matrix and has K positive entries σ_k , ($k = 1, \dots, K$) on its diagonal (i.e. positive singular values of $\ddot{\mathbf{P}}$).

⁴A unitary quaternion matrix $\ddot{\mathbf{B}} \in \mathbb{H}^{N \times N}$ has the following property: $\ddot{\mathbf{B}}\ddot{\mathbf{B}}^H = \ddot{\mathbf{B}}^H\ddot{\mathbf{B}} = \mathbf{I}_N$, with $\mathbf{I}_N \in \mathbb{R}^{N \times N}$ the identity matrix [28].

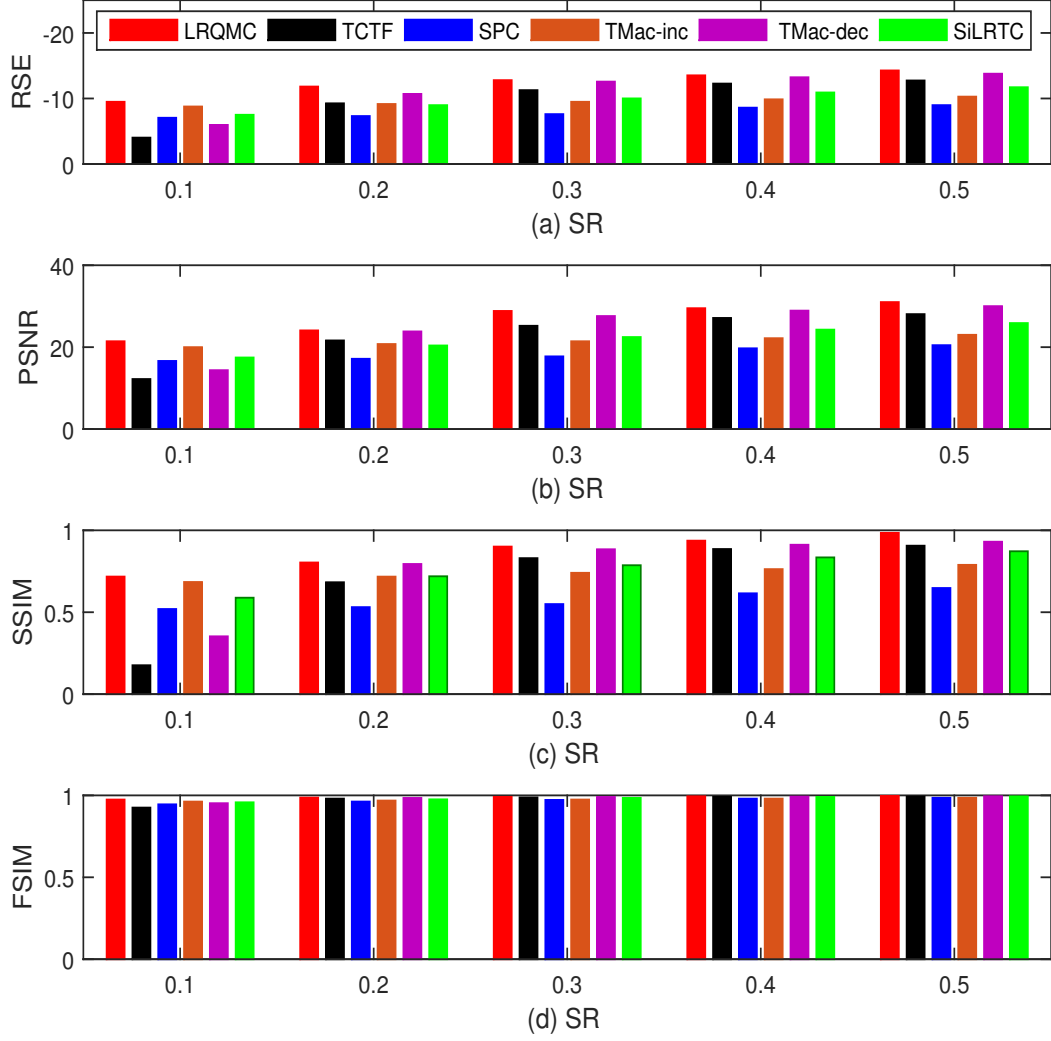


Figure 5: Comparison of RSE, PSNR, SSIM and FSIM results of different algorithms for color image recovery on Fig 4 (a) ($SR = [0.1, 0.2, 0.3, 0.4, 0.5]$). (a) RSE values, (b) PSNR values, (c) SSIM values, (d) FSIM values.

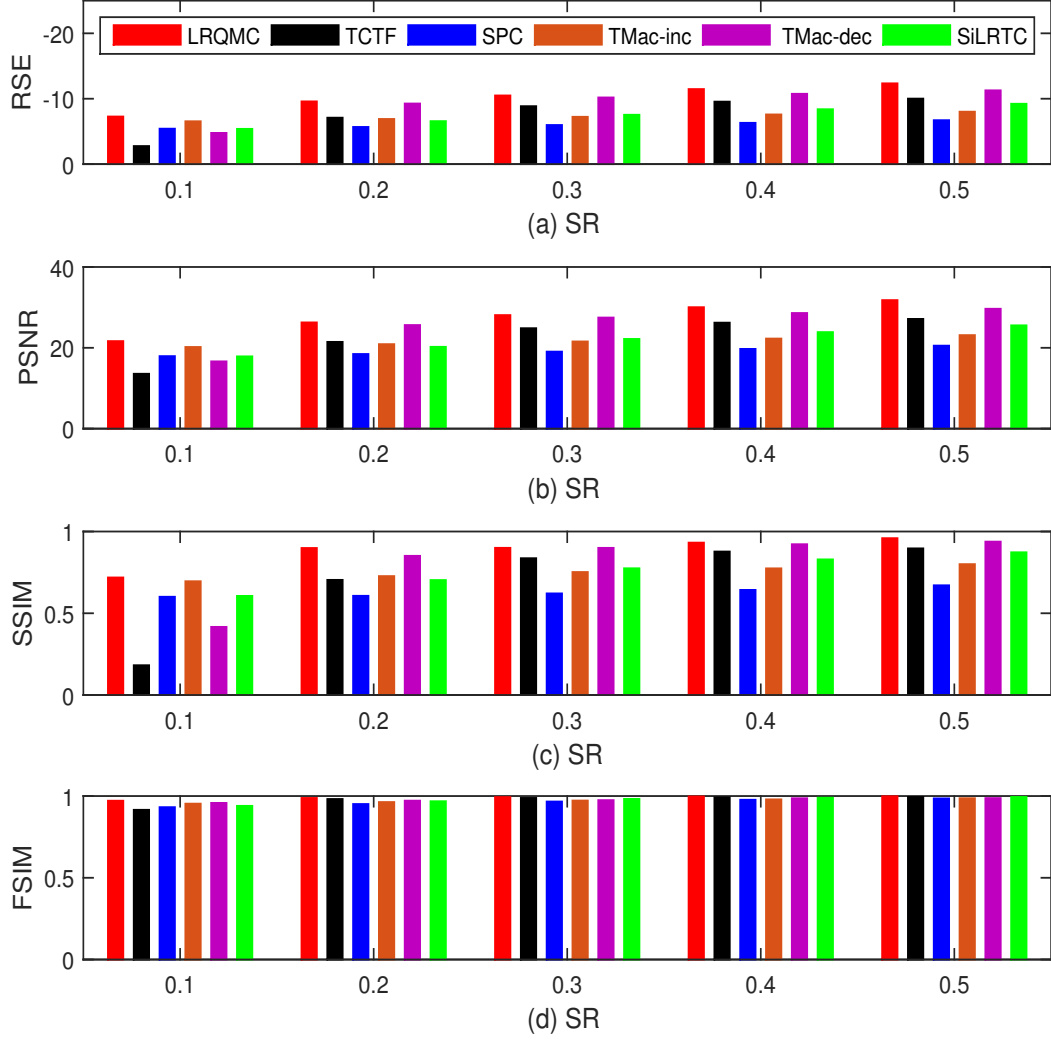


Figure 6: Comparison of RSE, PSNR, SSIM and FSIM results of different algorithms for color image recovery on Fig 4 (b) ($SR = [0.1, 0.2, 0.3, 0.4, 0.5]$). (a) RSE values, (b) PSNR values, (c) SSIM values, (d) FSIM values.

According to Lemma 1, we have

$$\begin{aligned} f(\ddot{\mathbf{A}})f(\ddot{\mathbf{P}})f(\ddot{\mathbf{B}}) &= f(\ddot{\mathbf{A}}\ddot{\mathbf{P}}\ddot{\mathbf{B}}) \\ &= \begin{pmatrix} \boldsymbol{\Sigma}_K & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}. \end{aligned}$$

Hence, from above, it is obvious that $\text{rank}(f(\ddot{\mathbf{P}})) = \text{rank}(f(\ddot{\mathbf{A}})f(\ddot{\mathbf{P}})f(\ddot{\mathbf{B}})) = 2K$, *i.e.*, $\text{rank}(\ddot{\mathbf{P}}) = \frac{1}{2}\text{rank}(f(\ddot{\mathbf{P}}))$ in Theorem 2 holds.

B Proof of the Theorem 3

The goal of this appendix is to prove the properties presented in Theorem 3.

Proof of (1): According to the QSVD in Lemma 1, there exist unitary quaternion matrices such that

$$\ddot{\mathbf{X}} = \ddot{\mathbf{A}} \begin{pmatrix} \boldsymbol{\Sigma}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \ddot{\mathbf{B}}.$$

We let

$$\ddot{\mathbf{A}} = \begin{pmatrix} \ddot{\mathbf{A}}_1 & \ddot{\mathbf{A}}_2 \end{pmatrix}, \quad \ddot{\mathbf{B}} = \begin{pmatrix} \ddot{\mathbf{B}}_1 \\ \ddot{\mathbf{B}}_2 \end{pmatrix},$$

where $\ddot{\mathbf{A}}_1 \in \mathbb{H}_K^{M \times K}$, $\ddot{\mathbf{A}}_2 \in \mathbb{H}_{(M-K)}^{M \times (M-K)}$, $\ddot{\mathbf{B}}_1 \in \mathbb{H}_K^{K \times N}$, $\ddot{\mathbf{B}}_2 \in \mathbb{H}_{(N-K)}^{(N-K) \times N}$. Then, we have

$$\begin{aligned} \ddot{\mathbf{X}} &= \begin{pmatrix} \ddot{\mathbf{A}}_1 & \ddot{\mathbf{A}}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{B}}_1 \\ \ddot{\mathbf{B}}_2 \end{pmatrix} \\ &= \ddot{\mathbf{A}}_1 \boldsymbol{\Sigma}_K \ddot{\mathbf{B}}_1 \\ &= \ddot{\mathbf{U}} \ddot{\mathbf{V}}, \end{aligned}$$

where $\ddot{\mathbf{U}} = \ddot{\mathbf{A}}_1 \in \mathbb{H}_K^{M \times K}$, $\ddot{\mathbf{V}} = \boldsymbol{\Sigma}_K \ddot{\mathbf{B}}_1 \in \mathbb{H}_K^{K \times N}$.

Proof of (2): Recall that if $\mathbf{P} \in \mathbb{C}^{M \times N}$ and $\mathbf{Q} \in \mathbb{C}^{N \times M}$ are two matrices, then we have $\text{rank}(\mathbf{P}\mathbf{Q}) \leq \min(\text{rank}(\mathbf{P}), \text{rank}(\mathbf{Q}))$. Thus, we immediately have

$$\begin{aligned} \text{rank}(\ddot{\mathbf{P}}\ddot{\mathbf{Q}}) &= \frac{1}{2}\text{rank}(f(\ddot{\mathbf{P}}\ddot{\mathbf{Q}})) \\ &= \frac{1}{2}\text{rank}(f(\ddot{\mathbf{P}})f(\ddot{\mathbf{Q}})) \\ &\leq \frac{1}{2}\min(\text{rank}(f(\ddot{\mathbf{P}})), \text{rank}(f(\ddot{\mathbf{Q}}))) \\ &= \min(\text{rank}(\ddot{\mathbf{P}}), \text{rank}(\ddot{\mathbf{Q}})). \end{aligned}$$

For property (3), actually, similar result and proof can be found in [42] (Lemma 5.1) and [43].

Acknowledgment

This work was supported by The Science and Technology Development Fund, Macau SAR (File no. FDCT/085/2018/A2).

References

- [1] R. H. Keshavan, A. Montanari, S. Oh, Matrix completion from a few entries, *IEEE Trans. Information Theory* 56 (6) (2010) 2980–2998. doi:10.1109/TIT.2010.2046205.
URL <https://doi.org/10.1109/TIT.2010.2046205>
- [2] Y. Hu, D. Zhang, J. Ye, X. Li, X. He, Fast and accurate matrix completion via truncated nuclear norm regularization, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (9) (2013) 2117–2130. doi:10.1109/TPAMI.2012.271.
URL <https://doi.org/10.1109/TPAMI.2012.271>
- [3] H. Wang, R. Zhao, Y. Cen, Rank adaptive atomic decomposition for low-rank matrix completion and its application on image recovery, *Neurocomputing* 145 (2014) 374–380. doi:10.1016/j.neucom.2014.05.021.
URL <https://doi.org/10.1016/j.neucom.2014.05.021>
- [4] X. Lin, G. Wei, Accelerated reweighted nuclear norm minimization algorithm for low rank matrix recovery, *Signal Processing* 114 (2015) 24–33. doi:10.1016/j.sigpro.2015.02.004.
URL <https://doi.org/10.1016/j.sigpro.2015.02.004>
- [5] Y. Yu, J. Peng, S. Yue, A new nonconvex approach to low-rank matrix completion with application to image inpainting, *Multidim. Syst. Sign. Process.* 30 (1) (2019) 145–174. doi:10.1007/s11045-018-0549-5.
URL <https://doi.org/10.1007/s11045-018-0549-5>
- [6] T. Yokota, Q. Zhao, A. Cichocki, Smooth PARAFAC decomposition for tensor completion, *IEEE Trans. Signal Processing* 64 (20) (2016) 5423–5436. doi:10.1109/TSP.2016.2586759.
URL <https://doi.org/10.1109/TSP.2016.2586759>
- [7] Q. Zhao, L. Zhang, A. Cichocki, Bayesian CP factorization of incomplete tensors with automatic rank determination, *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (9) (2015) 1751–1763. doi:10.1109/TPAMI.2015.2392756.
URL <https://doi.org/10.1109/TPAMI.2015.2392756>
- [8] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (1) (2013) 208–220. doi:10.1109/TPAMI.2012.39.
URL <https://doi.org/10.1109/TPAMI.2012.39>
- [9] J. A. Bengua, H. N. Phien, H. D. Tuan, M. N. Do, Efficient tensor completion for color image and video recovery: Low-rank tensor train, *IEEE Trans. Image Processing* 26 (5) (2017) 2466–2479. doi:10.1109/TIP.2017.2672439.
URL <https://doi.org/10.1109/TIP.2017.2672439>

- [10] P. Zhou, C. Lu, Z. Lin, C. Zhang, Tensor factorization for low-rank tensor completion, *IEEE Trans. Image Processing* 27 (3) (2018) 1152–1163. doi:10.1109/TIP.2017.2762595.
URL <https://doi.org/10.1109/TIP.2017.2762595>
- [11] Z. Long, Y. Liu, L. Chen, C. Zhu, Low rank tensor completion for multiway visual data, *Signal Processing* 155 (2019) 301–316. doi:10.1016/j.sigpro.2018.09.039.
URL <https://doi.org/10.1016/j.sigpro.2018.09.039>
- [12] Z. Zhang, S. Aeron, Exact tensor completion using t-svd, *IEEE Trans. Signal Processing* 65 (6) (2017) 1511–1526. doi:10.1109/TSP.2016.2639466.
URL <https://doi.org/10.1109/TSP.2016.2639466>
- [13] Z. Zhang, G. Ely, S. Aeron, N. Hao, M. E. Kilmer, Novel methods for multilinear data completion and de-noising based on tensor-svd, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23–28, 2014, 2014, pp. 3842–3849. doi:10.1109/CVPR.2014.485.
URL <https://doi.org/10.1109/CVPR.2014.485>
- [14] Y. Chen, C. Hsu, H. M. Liao, Simultaneous tensor decomposition and completion using factor priors, *IEEE Trans. Pattern Anal. Mach. Intell.* 36 (3) (2014) 577–591.
- [15] Soo-Chang Pei, Ching-Min Cheng, A novel block truncation coding of color images by using quaternion-moment-preserving principle, in: 1996 IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World. ISCAS 96, Vol. 2, 1996, pp. 684–687 vol.2. doi:10.1109/ISCAS.1996.541817.
- [16] B. Chen, Q. Liu, X. Sun, X. Li, H. Shu, Removing gaussian noise for colour images by quaternion representation and optimisation of weights in non-local means filter, *IET Image Processing* 8 (10) (2014) 591–600.
- [17] Ö. N. Subakan, B. C. Vemuri, A quaternion framework for color image smoothing and segmentation, *International Journal of Computer Vision* 91 (3) (2011) 233–250.
- [18] B. Chen, H. Shu, H. Zhang, G. Chen, C. Toumoulin, J. Dillenseger, L. Luo, Quaternion zernike moments and their invariants for color image analysis and object recognition, *Signal Processing* 92 (2) (2012) 308–318.
- [19] Y. Xu, L. Yu, H. Xu, H. Zhang, T. Nguyen, Vector sparse representation of color image using quaternion matrix analysis, *IEEE Trans. Image Processing* 24 (4) (2015) 1315–1329.
- [20] J. Xu, L. Ye, W. Luo, Color edge detection using multiscale quaternion convolution, *Int. J. Imaging Systems and Technology* 20 (4) (2010) 354–358.
- [21] S. Gai, G. Yang, M. Wan, L. Wang, Denoising color images by reduced quaternion matrix singular value decomposition, *Multidim. Syst. Sign. Process.* 26 (1) (2015) 307–320.
- [22] P. Bas, N. Le Bihan, J. . Chassery, Color image watermarking using quaternion fourier transform, in: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)., Vol. 3, 2003, pp. III–521. doi:10.1109/ICASSP.2003.1199526.

- [23] C. Zou, K. I. Kou, Y. Wang, Quaternion collaborative and sparse representation with application to color face recognition, *IEEE Trans. Image Processing* 25 (7) (2016) 3287–3302.
- [24] X. Han, J. Wu, L. Yan, L. Senhadji, H. Shu, Color image recovery via quaternion matrix completion, in: 2013 6th International Congress on Image and Signal Processing (CISP), Vol. 01, 2013, pp. 358–362.
- [25] K.-C. Toh, R. H. Ttnc, M. Todd, On the implementation and usage of sdpt3 a matlab software package for semidefinite-quadratic-linear programming, version 4.0, *Handbook on Semidefinite, Conic and Polynomial Optimization* 166. doi:10.1007/978146140769025.
- [26] J. F. Sturm, Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, *Optimization Methods and Software* 11. doi:10.1080/10556789908805766.
- [27] W. Rowan Hamilton, Ii. on quaternions; or on a new system of imaginaries in algebra, *Phil. Mag.*, 3rd Ser. 25. doi:10.1080/14786444408644923.
- [28] N. L. Bihan, J. I. Mars, Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing, *Signal Processing* 84 (7) (2004) 1177–1199.
- [29] F. ZHANG, Quaternions and matrices of quaternions, *Linear algebra abd its applications* 251 (1997) 21–57.
URL <https://ci.nii.ac.jp/naid/10029950538/en/>
- [30] P. R. Girard, *Quaternions, Clifford Algebras and Relativistic Physics*, 2007.
- [31] S. L. Altmann, Rotations, quaternions, and double groups, *Acta Crystallographica* 44 (4).
- [32] E. J. Cands, B. Recht, Exact matrix completion via convex optimization, *Foundations of Computational Mathematics* 9 (6) (2009) 717.
- [33] N. Gillis, F. Glineur, Low-rank matrix approximation with weights or missing data is np-hard, *SIAM J. Matrix Analysis Applications* 32 (4) (2011) 1149–1165.
- [34] S. Ma, D. Goldfarb, L. Chen, Fixed point and bregman iterative methods for matrix rank minimization, *Math. Program.* 128 (1-2) (2011) 321–353.
- [35] B. Recht, A simpler approach to matrix completion, *Journal of Machine Learning Research* 12 (2011) 3413–3430.
- [36] E. J. Candès, B. Recht, Exact matrix completion via convex optimization, *Foundations of Computational Mathematics* 9 (6) (2009) 717–772. doi:10.1007/s10208-009-9045-5.
URL <https://doi.org/10.1007/s10208-009-9045-5>
- [37] P. Jain, P. Netrapalli, S. Sanghavi, Low-rank matrix completion using alternating minimization, in: *STOC, ACM*, 2013, pp. 665–674.
- [38] Z. Wen, W. Yin, Y. Zhang, Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm, *Math. Program. Comput.* 4 (4) (2012) 333–361.

- [39] D. Xu, D. P. Mandic, The theory of quaternion matrix derivatives, *IEEE Trans. Signal Processing* 63 (6) (2015) 1543–1556.
- [40] Y. Xu, R. Hao, W. Yin, Z. Su, Parallel matrix factorization for low-rank tensor completion, *CoRR* abs/1312.1254.
- [41] E. J. Candes, B. Recht, Exact low-rank matrix completion via convex optimization, in: 2008 46th Annual Allerton Conference on Communication, Control, and Computing, 2008, pp. 806–812. doi:10.1109/ALLERTON.2008.4797640.
- [42] B. Recht, M. Fazel, P. A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, *SIAM Review* 52 (3) (2010) 471–501.
- [43] Y. Yang, M. Pesavento, S. Chatzinotas, B. E. Ottersten, Successive convex approximation algorithms for sparse signal estimation with nonconvex regularizations, *J. Sel. Topics Signal Processing* 12 (6) (2018) 1286–1302.