

Low-rank quaternion tensor completion for color video inpainting via a novel factorization strategy

Zhenzhi Qin*, Zhenyu Ming†, Defeng Sun‡, Liping Zhang*

March 26, 2024

Abstract

Recently, a quaternion tensor product named Qt-product was proposed, and then the singular value decomposition and the rank of a third-order quaternion tensor were given. From a more applicable perspective, we extend the Qt-product and propose a novel multiplication principle for third-order quaternion tensor named gQt-product. With the gQt-product, we introduce a brand-new singular value decomposition for third-order quaternion tensors named gQt-SVD and then define gQt-rank and multi-gQt-rank. We prove that the optimal low-rank approximation of a third-order quaternion tensor exists and some numerical experiments demonstrate the low-rankness of color videos. So, we apply the low-rank quaternion tensor completion to color video inpainting problems and present alternating least-square algorithms to solve the proposed low gQt-rank and multi-gQt-rank quaternion tensor completion models. The convergence analyses of the proposed algorithms are established and some numerical experiments on various color video datasets show the high recovery accuracy and computational efficiency of our methods.

keywords: quaternion tensor; color video inpainting; low-rank; singular value decomposition; tensor completion

1 Introduction

The purpose of this paper is to recover color videos via tensor completion. So far, tensors have been widely applied to signal processing [1], computer vision [2], graph analysis [3, 4, 5] and data mining [6], to name a few. Tensor decomposition is a fundamental tool to cope with large-scale data which is arranged in tensor-based forms, since the scale of tensor data can be notably reduced while most inherent information are still preserved by using decomposition techniques. Four commonly used tensor decomposition methods are CANDECOMP/PARAFAC (CP) decomposition [7, 8], Tucker decomposition [9], tensor singular value decomposition (t-SVD) [10] and Triple Decomposition [11], and the corresponding ranks are called CP rank [7, 8], Tucker rank [9], tubal rank [12] and Triple rank [11], respectively.

For a positive integer n , $[n] \doteq \{1, 2, \dots, n\}$. Suppose that $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_p}$ is an p -th order tensor, where $n_1, \dots, n_p \in \mathbb{N}^+$. The CP decomposition is to decompose \mathcal{A} as a sum of some outer products of p vectors:

$$\mathcal{A} = \sum_{i=1}^r a_1^{(i)} \circ a_2^{(i)} \circ \dots \circ a_p^{(i)}, \quad (1)$$

where the symbol “ \circ ” denotes the outer product and $a_j^{(i)} \in \mathbb{R}^{n_j}$, $i \in [r]$, $j \in [p]$. The smallest r required in CP decomposition (1) is defined as the CP rank of \mathcal{A} . It is learned from [13] that, in general, determining the CP rank of a given tensor whose order is no less than three is an NP-hard problem. In contrast to CP decomposition, Tucker decomposition is more computationally efficient. Hence a number of low-rank tensor completion and recovery models are based on Tucker rank [14, 15, 16]. Precisely, Tucker rank is a vector of the matrix ranks

$$\text{rank}_{\text{TC}}(\mathcal{A}) = (\text{rank}(\mathcal{A}_{(1)}), \text{rank}(\mathcal{A}_{(2)}), \dots, \text{rank}(\mathcal{A}_{(p)})),$$

*Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China (qzz19@mails.tsinghua.edu.cn; lipingzhang@mail.tsinghua.edu.cn)

†Theory Lab, Central Research Institute, 2012 Labs, Huawei Technologies Co., Ltd., Hong Kong (mathmzy@163.com)

‡Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (defeng.sun@polyu.edu.hk)

where $\mathcal{A}_{(i)} \in \mathbb{R}^{n_i \times (\prod_{k=1}^p n_k / n_i)}$ is mode- i matricization of tensor ($i \in [p]$). CP decomposition and Tucker decomposition are applicable to tensors with arbitrary orders. In 2011, Kilmer and Martin proposed a novel decomposition strategy specifically for third-order tensors [10]. Whereafter, the relevant tubal rank was introduced and studied in [12] and testified to have excellent performance for image and video inpainting problems [3].

Third-order tensors are the most widely used higher-order tensors in applications [3, 4, 5, 12, 17, 18]. For instance, a grey scale video can be viewed as a third-order tensor indexed by two spatial variables and one temporal variable. Unless otherwise specialized, tensors in this paper are of third-order. Low-rank tensor completion is one of the most important problems in tensor processing and analysis. It aims at filling in the missing entries of a partially observed low-rank tensor. Many practical datasets are highly structured in the sense that they can be approximately represented through a low-rank decomposition [19, 20]. As a consequence, the key idea of the recovery process is to find the low-rank approximation of the original tensor via the observed data, i.e.,

$$\min_{\mathcal{C}} \text{rank}(\mathcal{C}), \quad \text{s.t. } P_{\Omega}(\mathcal{C}) = P_{\Omega}(\mathcal{M}), \quad (2)$$

where $\text{rank}(\cdot)$ is a certain tensor rank and Ω is the index set locating the observed data, $P_{\Omega}(\cdot)$ is a linear operator that extracts the entries in Ω and fills the entries not in Ω with zeros, and \mathcal{M} is the raw tensor.

As previously mentioned, addressing model (2) with CP rank is an NP-hard problem. An alternative way is to employ Tucker rank instead of CP rank:

$$\min_{\mathcal{C}} \sum_{i=1}^p \text{rank}(\mathcal{C}_{(i)}), \quad \text{s.t. } P_{\Omega}(\mathcal{C}) = P_{\Omega}(\mathcal{M}), \quad (3)$$

and the nuclear-norm based convex relaxation of model (3) is considered as

$$\min_{\mathcal{C}} \sum_{i=1}^p \|\mathcal{C}_{(i)}\|_*, \quad \text{s.t. } P_{\Omega}(\mathcal{C}) = P_{\Omega}(\mathcal{M}). \quad (4)$$

However, Romera-Paredes et al. [21] proved that (4) is not a tight convex relaxation of (3), and SVD is needed to solve (4), which will lead to high computational cost when coping with large-scale issues. To overcome the computational difficulty, a matrix factorization method was designed by Xu et al. [19], which preserves the low-rank structure of the unfolded matrices, i.e.,

$$\min_{X^i, Y^i, \mathcal{C}} \sum_{i=1}^p \alpha_i \|X^i Y^i - \mathcal{C}_{(i)}\|_F^2, \quad \text{s.t. } P_{\Omega}(\mathcal{C}) = P_{\Omega}(\mathcal{M}), \quad (5)$$

where α_i is a positive weight parameter satisfying $\sum_{i=1}^p \alpha_i = 1$. A similar third-order tensor recovery method based on Triple decomposition is proposed in [11]. As pointed in [10, 12], directly unfolding a tensor will destroy the multi-way structure of the original data, resulting in vital information loss and degraded recovery performance. Besides, solving (5) requires to deal with p matrices and each matrix owns the same scale components as the original tensor. Thus the computational cost is relatively expensive. Instead, tubal rank has been adopted in (2) and testified to have not only promising recovery performance but also efficient computational process. Semerci et al. [22] developed a new tensor nuclear norm (TNN) based on t-SVD, and subsequently Zhang et al. [4] applied TNN to tensor completion problems. Zhou et al. [3] proposed the following model based on tubal rank and tensor product (t-product) to replace model (2):

$$\min_{\mathcal{A}, \mathcal{B}, \mathcal{C}} \frac{1}{2} \|\mathcal{A} * \mathcal{B} - \mathcal{C}\|_F^2, \quad \text{s.t. } P_{\Omega}(\mathcal{C}) = P_{\Omega}(\mathcal{M}), \quad (6)$$

where “*” denotes the t-product [10]. From [10, 12], one can deal with t-product via fast Fourier transform (FFT) and block diagonalization of third-order tensors, which can significantly reduce the computational cost.

Motivation. We now briefly describe the motivation of this paper here. All the above methods explore the approximate low-rank property of higher-order tensors. However, they are not good enough for the classic color video inpainting problems. In specific, the traditional t-SVD and Triple decomposition are specially designed for third-order tensors, while a color video can be naturally described as a fourth-order

tensor, with four dimensions representing the length, width, frame numbers and RGB-channels of the considered color video, respectively. Moreover, the computational complexity of CP decomposition is NP-hard, and the unfolding operation in Tucker decomposition will destroy the original multi-way structure of the data. Therefore, it is desirable to design a new type of tensor factorization strategy which can tackle the above issues in terms of the capability, the recovery performance and the computational cost. Notice that the red, green and blue channel pixel values can be intuitively encoded on the three imaginary parts of a quaternion. The use of quaternion matrices for color image representation has been fully studied in the literature [23, 24, 25, 26, 27]. In 2022, we proposed a quaternion tensor product (Qt-product) and then introduced the singular value decomposition (Qt-SVD) and the rank of a third-order quaternion tensor (Qt-rank) by employing the discrete Fourier transformation (DFT) technique. They also proved that the existence of the best low-rank approximation of a third order quaternion tensor from the theoretical point and the low-rank of color videos from numerical experiments [28]. But this is not applied to solve color video inpainting problems, and the DFT used in [28] is very special. From a more applicable perspective, in this paper, we generalize the Qt-product and propose a novel multiplication principle for third-order quaternion tensor, and then establish low-rank quaternion tensor completion models to recover color videos.

Contribution. By introducing an extensive quaternion discrete Fourier transformation (QDFT) based on a pure quaternion basis, we propose a novel multiplication principle for third-order quaternion tensor named gQt-product, and then a new SVD is given. With such SVD, we establish two low-rank quaternion tensor completion models to recover the incomplete color video data, and present an alternating least-squared (ALS) algorithm to solve the color video inpainting problems. The numerical experiments show that our methods outperform other state-of-the-arts in the recovery accuracy and computational efficiency. The main contributions are summarized as follows.

- A generalized QDFT based on a pure quaternion basis is introduced and a novel quaternion tensor product named gQt-product is proposed. With the gQt-product, we define identity quaternion tensor, unitary quaternion tensor, conjugate transpose and inverse of quaternion tensor. We prove that the collection of all invertible $n \times n \times l$ quaternion tensors forms a ring under standard tensor addition and the gQt-product.
- A new gQt-product based SVD for quaternion tensors named gQt-SVD is given, and then the gQt-rank and the nuclear norm of third-order quaternion tensor are defined. We prove that the optimal low-rank approximation of third-order quaternion tensor exists and some numerical experiments demonstrate the low-rankness of color videos. Note that gQt-rank is only defined on one mode of third order quaternion tensor without low rank structure in the other two modes, so we also introduce multi-gQt-rank.
- To cope with color video inpainting problem, we construct low-rank quaternion tensor completion models (2) based on gQt-rank and multi-gQt-rank, and further propose their evolved forms (28) and (59) via the gQt-product. We present an ALS algorithm to solve (28) and (59), and also show that the sequence generated by the ALS algorithm globally converges to a stationary point of the problem by using the Kurdyka–Łojasiewicz property exhibited in the resulting problem. Extensive numerical experiments on various color video datasets show the high recovery accuracy and computational efficiency of our methods. Especially, the criterion of the recovery performance illustrates our gQt-SVD-based method is superior to the commonly used t-SVD-based one.

The rest of this paper is organized as follows. In Section 2, we list some existing results for quaternion matrices and quaternion tensors. and discuss the Fourier transform of quaternion tensors. In Section 3, we introduce a generalized QDFT based on a pure quaternion basis, and then gQt-product, gQt-SVD, gQt-rank and multi-gQt-rank are defined. In Section 4, we establish related low-rank quaternion tensor completion models to recover the incomplete color video data, and present the ALS algorithm to solve the resulting problem. Moreover, its convergence rate analysis is also established. In Section 5, some numerical results are reported to confirm the advantages of gQt-SVD-based methods. The conclusions are drawn in Section 6.

2 Preliminary

2.1 Quaternions

Let \mathbb{R} and \mathbb{C} denote the real field and the complex field, respectively. The quaternion field, denoted as \mathbb{H} , is a four-dimensional vector space over real number field \mathbb{R} with an ordered basis, denoted by $\mathbf{1}$, \mathbf{i} , \mathbf{j} and \mathbf{k} . Here \mathbf{i} , \mathbf{j} and \mathbf{k} are three imaginary units with the following multiplication laws:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1, \mathbf{ij} = -\mathbf{ji} = \mathbf{k}, \mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \mathbf{ki} = -\mathbf{ik} = \mathbf{j}.$$

Let $x = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \in \mathbb{H}$, where $a, b, c, d \in \mathbb{R}$, then the conjugate of x is defined by

$$x^* \doteq a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k},$$

the norm of x is

$$|x| = |x^*| = \sqrt{xx^*} = \sqrt{x^*x} = \sqrt{a^2 + b^2 + c^2 + d^2}.$$

and if $x \neq 0$, then $x^{-1} = \frac{x^*}{|x|^2}$.

2.2 Quaternion matrix and quaternion tensor

We give some notations here. Scalars, vectors, matrices and third-order tensors are denoted as lowercase letters (a, b, \dots), bold-case lowercase letters ($\mathbf{a}, \mathbf{b}, \dots$), capital letters (A, B, \dots) and Euler script letters ($\mathcal{A}, \mathcal{B}, \dots$), respectively. We use $\mathbf{0}$, O and \mathcal{O} to denote zero vector, zero matrix and zero tensor with appropriate dimensions. We use symbols e to represent the vector whose elements are all 1, and I and \mathcal{I} to denote the identity matrix, and the identity tensor, respectively. The identity tensor \mathcal{I} will be defined in Section 3.

Then a quaternion matrix $A = (A_{ij}) \in \mathbb{H}^{n_1 \times n_2}$ can be denoted as

$$A = A_e + A_i\mathbf{i} + A_j\mathbf{j} + A_k\mathbf{k},$$

where $A_e, A_i, A_j, A_k \in \mathbb{R}^{n_1 \times n_2}$. The transpose of A is $A^T = (A_{ji})$. The conjugate transpose of A is

$$A^* = (A_{ji}^*) = A_e^T - A_i^T\mathbf{i} - A_j^T\mathbf{j} - A_k^T\mathbf{k}.$$

The Frobenius norm of A is

$$\|A\|_F \doteq \sqrt{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} |A_{ij}|^2}.$$

With a simple calculation, it can be seen that

$$\|A\|_F^2 = \text{tr}(AA^*) = \text{tr}(A^*A), \quad (7)$$

where $\text{tr}(\cdot)$ is the trace of a matrix.

Let $A \in \mathbb{H}^{n \times n}$. A is a unitary matrix if and only if $AA^* = A^*A = I_n$, where $I_n \in \mathbb{R}^{n \times n}$ is the real $n \times n$ identity matrix. A is invertible if $AB = BA = I_n$ for some $B \in \mathbb{H}^{n \times n}$. The following lemma gives Some properties of invertible quaternion matrix which can be found in [29, Theorem 4.1].

Lemma 2.1. *Let $A \in \mathbb{H}^{n_1 \times n_2}$ and $B \in \mathbb{H}^{n_2 \times n_3}$, then*

- (i) $(AB)^* = B^*A^*$,
- (ii) $(AB)^{-1} = B^{-1}A^{-1}$ if A and B are invertible,
- (iii) $(A^*)^{-1} = (A^{-1})^*$ if A is invertible.

The following theorem for the SVD of quaternion matrix (QSVD) was given in [29].

Theorem 2.1. *Any quaternion matrix $A \in \mathbb{H}^{n_1 \times n_2}$ has the following QSVD form*

$$A = U \begin{bmatrix} \Sigma_r & O \\ O & O \end{bmatrix} V^*,$$

where $U \in \mathbb{H}^{n_1 \times n_1}$ and $V \in \mathbb{H}^{n_2 \times n_2}$ are unitary, and $\Sigma_r = \text{diag}\{\sigma_1, \dots, \sigma_r\}$ is a real nonnegative diagonal matrix, with $\sigma_1 \geq \dots \geq \sigma_r$ as the singular values of A .

By Theorem 2.1, the nuclear norm of A is defined as $\|A\|_* = \sum_{i=1}^r \sigma_i$. The quaternion rank of A is the number of its singular values, denoted as $\text{rank}(A)$. It follows from [30, Lemma 9] that we can prove the following lemma, which reveals the relationship between the matrix-factorization and the nuclear norm of a quaternion matrix.

Lemma 2.2. *For a given quaternion matrix $A \in \mathbb{H}^{n_1 \times n_2}$,*

$$\|A\|_* = \min_{A=XY^*} \|X\|_F \|Y\|_F = \min_{A=XY^*} \frac{1}{2} (\|X\|_F^2 + \|Y\|_F^2) \quad (8)$$

Proof. We denote three parts in (8) as (i), (ii) and (iii) from left to right.

(ii) \leq (iii): This follows from the arithmetic mean and geometric mean inequality.

(iii) \leq (i): We decompose A into the form in Theorem 2.1, and then set

$$\tilde{X} = U \begin{bmatrix} \Sigma_r^{1/2} & O \\ O & O \end{bmatrix}, \quad \tilde{Y} = \begin{bmatrix} \Sigma_r^{1/2} & O \\ O & O \end{bmatrix} V.$$

Hence, \tilde{X}, \tilde{Y} are feasible matrices of (iii), and $\frac{1}{2}(\|\tilde{X}\|_F^2 + \|\tilde{Y}\|_F^2) = \|A\|_*$, which implies (iii) \leq (i).

(i) \leq (ii): For all X, Y with $A = XY^*$, let $\mathbf{u}_i, \mathbf{v}_i$ be the i -th column of U and V , respectively. Then

$$\begin{aligned} \|A\|_* &= \sum_{i=1}^r \mathbf{u}_i^* A \mathbf{v}_i = \sum_{i=1}^r (X^* \mathbf{u}_i)^* (Y^* \mathbf{v}_i) \\ &\leq \sum_{i=1}^r \|X^* \mathbf{u}_i\|_F \|Y^* \mathbf{v}_i\|_F \\ &\leq \left(\sum_{i=1}^r \|X^* \mathbf{u}_i\|_F^2 \right)^{1/2} \left(\sum_{i=1}^r \|Y^* \mathbf{v}_i\|_F^2 \right)^{1/2} \\ &\leq \|X^* U\|_F \|Y^* V\|_F = \|X\|_F \|Y\|_F, \end{aligned}$$

where the first and second inequalities are from Cauchy-Schwarz inequality, which can be verified on quaternion, and the third inequality holds because we complete the rest part of U and V . \square

A third-order quaternion tensor $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ is expressed as

$$\mathcal{A} = (\mathcal{A}_{i_1 i_2 i_3}), \quad \mathcal{A}_{i_1 i_2 i_3} \in \mathbb{H}, \quad 1 \leq i_t \leq n_t, \quad 1 \leq t \leq 3.$$

Also, it can be expressed as

$$\mathcal{A} = \mathcal{A}_e + \mathcal{A}_i \mathbf{i} + \mathcal{A}_j \mathbf{j} + \mathcal{A}_k \mathbf{k}. \quad (9)$$

where $\mathcal{A}_e, \mathcal{A}_i, \mathcal{A}_j, \mathcal{A}_k \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. We use the Matlab notations $\mathcal{A}(i, :, :)$, $\mathcal{A}(:, j, :)$ and $\mathcal{A}(:, :, k)$ to denote its i -th horizontal, j -th lateral and k -th frontal slice, respectively. Let $A^{(k)} = \mathcal{A}(:, :, k)$ be the k -th ($k \in [n_3]$) frontal slice and \mathcal{A}^* denote its conjugate transpose (see Section 3). The Frobenius norm of \mathcal{A} is the sum of all norms of its entries, i.e.,

$$\|\mathcal{A}\|_F \doteq \sqrt{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} |\mathcal{A}_{ijk}|^2}.$$

The block circulant matrix $\text{circ}(\mathcal{A}) \in \mathbb{H}^{n_1 n_3 \times n_2 n_3}$ of \mathcal{A} is defined as

$$\text{circ}(\mathcal{A}) \doteq \begin{bmatrix} A^{(1)} & A^{(n_3)} & \dots & A^{(2)} \\ A^{(2)} & A^{(1)} & \dots & A^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ A^{(n_3)} & A^{(n_3-1)} & \dots & A^{(1)} \end{bmatrix}.$$

The operator “unfold” is defined as

$$\text{unfold}(\mathcal{A}) \doteq [A^{(1)}; A^{(2)}; \dots; A^{(n_3)}],$$

and its inverse operator “fold” is defined by $\text{fold}(\text{unfold}(\mathcal{A})) = \mathcal{A}$. The operator “diag” of \mathcal{A} is given as

$$\text{diag}(\mathcal{A}) \doteq \begin{bmatrix} A^{(1)} & & & \\ & A^{(2)} & & \\ & & \ddots & \\ & & & A^{(n_3)} \end{bmatrix}.$$

In 2022, Qin et al [28] noted that a quaternion $x = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ can be written as $x = (a + b\mathbf{i}) + \mathbf{j}(c - d\mathbf{i})$ and then the quaternion tensor \mathcal{A} can be written as the form of $\mathcal{A} = \mathcal{A}_{1,\mathbf{i}} + \mathbf{j}\mathcal{A}_{\mathbf{j},\mathbf{k}}$ with $\mathcal{A}_{1,\mathbf{i}}, \mathcal{A}_{\mathbf{j},\mathbf{k}} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$. So, Qin et al [28] defined the following quaternion tensor product named Qt-product, and they also give SVD of third-order quaternion tensor by using the Qt-product and the following DFT.

Definition 2.1 (Qt-product [28]). For $\mathcal{A} \in \mathbb{H}^{n_1 \times r \times n_3}$ and $\mathcal{B} \in \mathbb{H}^{r \times n_2 \times n_3}$, define

$$\mathcal{A} *_{\mathbb{H}} \mathcal{B} \doteq \text{fold}((\text{circ}(\mathcal{A}_{1,\mathbf{i}}) + \mathbf{j}\text{circ}(\mathcal{A}_{\mathbf{j},\mathbf{k}}) \cdot (P_{n_3} \otimes I_r)) \cdot \text{unfold}(\mathcal{B})) \in \mathbb{H}^{n_1 \times n_2 \times n_3},$$

where the symbol “ \otimes ” means the Kronecker product, the matrix $P_{n_3} = (P_{ij}) \in \mathbb{R}^{n_3 \times n_3}$ is a permutation matrix with $P_{11} = P_{ij} = 1$ if $i + j = n_3 + 2$, $2 \leq i, j \leq n_3$; $P_{ij} = 0$, otherwise.

The DFT used in [28] is given as the form of the normalized DFT matrix $F_{n_3} \in \mathbb{C}^{n_3 \times n_3}$ with

$$F_{n_3}(i, j) = \frac{1}{\sqrt{n_3}} \omega^{(i-1)(j-1)}, \quad \omega = \exp(-2\pi\mathbf{i}/n_3) \text{ and } i, j \in [n_3]. \quad (10)$$

The DFT (10) plays an important role in the Qt-SVD of third-order quaternion tensor given in [28]. As a generalization of the traditional Fourier transform, the quaternion Fourier transform was first defined by Ell [31] to process quaternion signal. Motivated by the idea of quaternion Fourier transform in [31] and from a more applicable perspective, in this paper we define a new quaternion DFT (QDFT) based on a unit pure quaternion $\mu = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$ with $\mu^2 = -1$, which can be regards as a generalization of DFT (10). And then, we introduce a novel multiplication principle for third-order quaternion tensor via the new defined QDFT.

3 QDFT, gQt-Product and gQt-SVD

One major contribution of this paper is to introduce a novel multiplication principle for third-order quaternion tensor, named gQt-product, via our new defined QDFT. The gQt-product is also the cornerstone of the quaternion tensor decomposition. In this section, we will introduce gQt-product and some relevant properties. Theorem 3.1 is one of the main results, which shows the relationship between gQt-product of quaternion tensors and matrix product of their QDFT matrices. Furthermore, we propose a new SVD of quaternion tensor via gQt-product, named gQt-SVD. With gQt-SVD, we can find the low-rank optimal approximation of quaternion tensor.

3.1 QDFT: a new DFT of third-order quaternion tensor

In this subsection, we define a new quaternion DFT (QDFT) based on a unit pure quaternion $\mu = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$ with $\mu^2 = -1$, which can be regards as a generalization of DFT (10). Because the quaternion multiplication is not commutative, QDFT of vectors can be defined by the sum of components multiplied by the exponential kernel of the transform from the right or from the left.

Let $F_{\mu, n_3} \in \mathbb{H}^{n_3 \times n_3}$ be the normalized QDFT matrix with the (i, j) -th element as

$$F_{\mu, n_3}(i, j) = \frac{1}{\sqrt{n_3}} \omega^{(i-1)(j-1)}, \quad i, j \in [n_3], \quad (11)$$

where the kernel ω is defined as

$$\omega = \exp(-\mu 2\pi/n_3) = \cos(2\pi/n_3) - \mu \sin(2\pi/n_3).$$

Obviously, it follows that

$$\omega^p = \exp(-\mu 2\pi p/n_3) = \cos(2\pi p/n_3) - \mu \sin(2\pi p/n_3), \quad p \in \mathbb{Z}. \quad (12)$$

Clearly, when $\mu = \mathbf{i}$, QDFT matrix defined as (11) is equal to DFT matrix given in (10). So, QDFT is more applicable than DFT.

It is easy to see that the result of QDFT of $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ is still a quaternion tensor $\hat{\mathcal{A}} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ with

$$\hat{\mathcal{A}}(i, j, :) = \sqrt{n_3} F_{\mu, n_3} \mathcal{A}(i, j, :), \quad i \in [n_1], j \in [n_2].$$

Moreover,

$$\|\hat{\mathcal{A}}\|_F^2 = n_3 \|\mathcal{A}\|_F^2. \quad (13)$$

We also denote the QDFT of \mathcal{A} as $\text{fft}(\mathcal{A})$, and its inverse operator “ifft” is defined by $\text{ifft}(\text{fft}(\mathcal{A})) = \mathcal{A}$. It is known that any real circulant matrix can be diagonalized by the normalized DFT matrix [32]. For QDFT, by simple calculation, we also obtain the same result for any real tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$:

$$(F_{\mu, n_3} \otimes I_{n_1}) \cdot \text{circ}(\mathcal{A}) \cdot (F_{\mu, n_3}^* \otimes I_{n_2}) = \text{diag}(\hat{\mathcal{A}}). \quad (14)$$

Denote $P_n = (P_{ij}) \in \mathbb{R}^{n \times n}$ as a permutation matrix with $P_{11} = P_{ij} = 1$ if $i + j = n + 2$, $2 \leq i, j \leq n$; $P_{ij} = 0$, otherwise. We now give some special kernels of QDFT matrices, which will be used to introduce gQt-product. For a unit pure quaternion $\mu = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$ with $\mu^2 = -1$, set

$$\mu_{\mathbf{i}} = a\mathbf{i} - b\mathbf{j} - c\mathbf{k}, \quad \mu_{\mathbf{j}} = -a\mathbf{i} + b\mathbf{j} - c\mathbf{k}, \quad \mu_{\mathbf{k}} = -a\mathbf{i} - b\mathbf{j} + c\mathbf{k}. \quad (15)$$

The following lemma gives the relationship between QDFT matrices in the form of (11) with different quaternion basis.

Lemma 3.1. *For two unit pure quaternion numbers μ_1 and μ_2 with $\mu_1^2 = \mu_2^2 = -1$, it holds that*

$$F_{\mu_1, n}^* F_{\mu_2, n} = \frac{1}{2}(1 - \mu_1 \mu_2) I_n + \frac{1}{2}(1 + \mu_1 \mu_2) P_n, \quad F_{\mu_1, n} F_{\mu_2, n} = \frac{1}{2}(1 + \mu_1 \mu_2) I_n + \frac{1}{2}(1 - \mu_1 \mu_2) P_n.$$

Proof. It follows from (11) and (12) that

$$F_{\mu_t, n} = C_n - \mu_t S_n, \quad F_{\mu_t, n}^* = C_n + \mu_t S_n, \quad t = 1, 2, \quad (16)$$

where $C_n = (C_{ij}) \in \mathbb{R}^{n \times n}$ and $S_n = (S_{ij}) \in \mathbb{R}^{n \times n}$ satisfy

$$C_{ij} = \frac{1}{\sqrt{n}} \cos(2\pi(i-1)(j-1)/n), \quad S_{ij} = \frac{1}{\sqrt{n}} \sin(2\pi(i-1)(j-1)/n).$$

By (10) and (11), the traditional DFT matrix F_n can be written as $F_{\mathbf{i}, n}$. By simple computation, we have

$$F_{\mathbf{i}, n}^* F_{\mathbf{i}, n} = I_n, \quad F_{\mathbf{i}, n} F_{\mathbf{i}, n} = P_n.$$

Combining (16), it is easy to get

$$C_n^2 + S_n^2 = I_n, \quad C_n^2 - S_n^2 = P, \quad C_n S_n = S_n C_n = O.$$

Hence,

$$\begin{aligned} F_{\mu_1, n}^* F_{\mu_2, n} &= (C_n + \mu_1 S_n)(C_n - \mu_2 S_n) = C_n^2 - \mu_1 \mu_2 S_n^2 \\ &= \frac{1}{2}(1 - \mu_1 \mu_2) I_n + \frac{1}{2}(1 + \mu_1 \mu_2) P_n. \end{aligned}$$

In the same way, we can prove that $F_{\mu_1, n} F_{\mu_2, n} = \frac{1}{2}(1 + \mu_1 \mu_2) I_n + \frac{1}{2}(1 - \mu_1 \mu_2) P_n$. \square

3.2 gQt-product: a new product between third-order quaternion tensors

In this subsection, based on QDFT (11) and the form of quaternion tensor \mathcal{A} as (9), we introduce the concept of gQt-product and define identity quaternion tensor and inverse quaternion tensor. Moreover, the relation of gQt-product of quaternion tensors and matrix product of their QDFT matrices is given.

Definition 3.1 (gQt-product). For any given unit pure quaternion $\mu = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$ with $\mu^2 = -1$, let $\mu_{\mathbf{i}}, \mu_{\mathbf{j}}, \mu_{\mathbf{k}}$ be given as (15) and matrices $T_{\mathbf{i}}, T_{\mathbf{j}}, T_{\mathbf{k}}$ be defined as

$$T_{\mathbf{i}} = F_{\mu_{\mathbf{i}}, n_3}^* F_{\mu, n_3}, \quad T_{\mathbf{j}} = F_{\mu_{\mathbf{j}}, n_3}^* F_{\mu, n_3}, \quad T_{\mathbf{k}} = F_{\mu_{\mathbf{k}}, n_3}^* F_{\mu, n_3}.$$

Define gQt-product of $\mathcal{A} \in \mathbb{H}^{n_1 \times r \times n_3}$ and $\mathcal{B} \in \mathbb{H}^{r \times n_2 \times n_3}$ as

$$\mathcal{A} *_{\mu} \mathcal{B} \doteq \text{fold}((\text{circ}(\mathcal{A}_{\mathbf{e}}) + \mathbf{i}\text{circ}(\mathcal{A}_{\mathbf{i}}) \cdot (T_{\mathbf{i}} \otimes I_r) + \mathbf{j}\text{circ}(\mathcal{A}_{\mathbf{j}}) \cdot (T_{\mathbf{j}} \otimes I_r) + \mathbf{k}\text{circ}(\mathcal{A}_{\mathbf{k}}) \cdot (T_{\mathbf{k}} \otimes I_r)) \cdot \text{unfold}(\mathcal{B})).$$

Clearly, $\mathcal{A} *_{\mu} \mathcal{B} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$.

We have the following remarks for Definition 3.1.

Remark 3.1. When $\mu = \mathbf{i}$, gQt-product “ $*_{\mu}$ ” becomes Qt-product “ $*_{\mathbb{H}}$ ” in Definition 2.1 [28]. Moreover, if we restrict the product to real tensors, gQt-product is just t-product defined in [10]. In this view, gQt-product is a generalization of t-product in a wider field.

Remark 3.2. For any unit pure quaternion $\mu = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$ with $a^2 + b^2 + c^2 = 1$, by Lemma 3.1, we can get

$$\begin{aligned} F_{\mu_{\mathbf{i}}, n_3}^* F_{\mu, n_3} &= (a^2 - a(b\mathbf{k} - c\mathbf{j}))I_{n_3} + (1 - a^2 + a(b\mathbf{k} - c\mathbf{j}))P_{n_3}, \\ F_{\mu_{\mathbf{j}}, n_3}^* F_{\mu, n_3} &= (b^2 - b(c\mathbf{i} - a\mathbf{k}))I_{n_3} + (1 - b^2 + b(c\mathbf{i} - a\mathbf{k}))P_{n_3}, \\ F_{\mu_{\mathbf{k}}, n_3}^* F_{\mu, n_3} &= (c^2 - c(a\mathbf{j} - b\mathbf{i}))I_{n_3} + (1 - c^2 + c(a\mathbf{j} - b\mathbf{i}))P_{n_3}. \end{aligned}$$

Hence, it is easy to see that $T_{\mathbf{i}}, T_{\mathbf{j}}, T_{\mathbf{k}}$ is the combination of identity matrix I_{n_3} and permutation matrix P_{n_3} , with the sum of the coefficients being one. In this vision, the operator from $\text{circ}(\mathcal{A})$ to $\text{circ}(\mathcal{A}_{\mathbf{e}}) + \mathbf{i}\text{circ}(\mathcal{A}_{\mathbf{i}}) \cdot (T_{\mathbf{i}} \otimes I_r) + \mathbf{j}\text{circ}(\mathcal{A}_{\mathbf{j}}) \cdot (T_{\mathbf{j}} \otimes I_r) + \mathbf{k}\text{circ}(\mathcal{A}_{\mathbf{k}}) \cdot (T_{\mathbf{k}} \otimes I_r)$ will not change the magnitude, so there is no loss of information in gQt-product.

We now present the relation of gQt-product of quaternion tensors and matrix product of their QDFT matrices in the following theorem. For any given $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$, setting its QDFT tensor as $\hat{\mathcal{A}} = \text{fft}(\mathcal{A})$, then we have

$$\text{unfold}(\hat{\mathcal{A}}) = \sqrt{n_3}(F_{\mu, n_3} \otimes I_{n_1}) \cdot \text{unfold}(\mathcal{A}). \quad (17)$$

Theorem 3.1. Let $\mathcal{A} \in \mathbb{H}^{n_1 \times r \times n_3}$, $\mathcal{B} \in \mathbb{H}^{r \times n_2 \times n_3}$ and $\mathcal{C} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$, $\hat{\mathcal{A}}, \hat{\mathcal{B}}, \hat{\mathcal{C}}$ be their QDFT tensors. Then,

$$\text{diag}(\hat{\mathcal{C}}) = \text{diag}(\hat{\mathcal{A}}) \cdot \text{diag}(\hat{\mathcal{B}}) \Leftrightarrow \mathcal{C} = \mathcal{A} *_{\mu} \mathcal{B}.$$

Proof. Clearly, $\mathbf{i}F_{\mu, n_3} = F_{\mu_{\mathbf{i}}, n_3}\mathbf{i}$. Thus, it follows from (17) and (14) that

$$\begin{aligned} \text{diag}(\text{fft}(\mathcal{A}_{\mathbf{i}}\mathbf{i})) &= \text{diag}\left(\text{fold}(\sqrt{n_3}(F_{\mu, n_3} \otimes I_{n_1}) \cdot \text{unfold}(\mathcal{A}_{\mathbf{i}}\mathbf{i}))\right) \\ &= \text{diag}\left(\text{fold}(\sqrt{n_3}(F_{\mu, n_3} \otimes I_{n_1}) \cdot \text{unfold}(\mathcal{A}_{\mathbf{i}}))\right)\mathbf{i} \\ &= \text{diag}(\text{fft}(\mathcal{A}_{\mathbf{i}}))\mathbf{i} \\ &= (F_{\mu, n_3} \otimes I_{n_1}) \cdot \text{circ}(\mathcal{A}_{\mathbf{i}}) \cdot (F_{\mu_{\mathbf{i}}, n_3}^* \otimes I_{n_2})\mathbf{i} \\ &= (F_{\mu, n_3} \otimes I_{n_1}) \cdot \mathbf{i}\text{circ}(\mathcal{A}_{\mathbf{i}}) \cdot (F_{\mu_{\mathbf{i}}, n_3}^* \otimes I_{n_2}), \end{aligned} \quad (18)$$

where the second equality holds due to the homogeneity of operators “fold”, “unfold” and “diag”. Similarly, we also have

$$\text{diag}(\text{fft}(\mathcal{A}_{\mathbf{j}}\mathbf{j})) = (F_{\mu, n_3} \otimes I_{n_1}) \cdot \mathbf{j}\text{circ}(\mathcal{A}_{\mathbf{j}}) \cdot (F_{\mu_{\mathbf{j}}, n_3}^* \otimes I_{n_2}), \quad (19)$$

$$\text{diag}(\text{fft}(\mathcal{A}_{\mathbf{k}}\mathbf{k})) = (F_{\mu, n_3} \otimes I_{n_1}) \cdot \mathbf{k}\text{circ}(\mathcal{A}_{\mathbf{k}}) \cdot (F_{\mu_{\mathbf{k}}, n_3}^* \otimes I_{n_2}). \quad (20)$$

So, by (18), (19) and (20), it holds that

$$\begin{aligned} \text{diag}(\hat{\mathcal{A}}) &= \text{diag}(\text{fft}(\mathcal{A}_{\mathbf{e}})) + \text{diag}(\text{fft}(\mathcal{A}_{\mathbf{i}}\mathbf{i})) + \text{diag}(\text{fft}(\mathcal{A}_{\mathbf{j}}\mathbf{j})) + \text{diag}(\text{fft}(\mathcal{A}_{\mathbf{k}}\mathbf{k})) \\ &= (F_{\mu, n_3} \otimes I_{n_1}) \cdot (\text{circ}(\mathcal{A}_{\mathbf{e}}) \cdot (F_{\mu, n_3}^* \otimes I_{n_2}) + \mathbf{i}\text{circ}(\mathcal{A}_{\mathbf{i}}) \cdot (F_{\mu_{\mathbf{i}}, n_3}^* \otimes I_{n_2}) \\ &\quad + \mathbf{j}\text{circ}(\mathcal{A}_{\mathbf{j}}) \cdot (F_{\mu_{\mathbf{j}}, n_3}^* \otimes I_{n_2}) + \mathbf{k}\text{circ}(\mathcal{A}_{\mathbf{k}}) \cdot (F_{\mu_{\mathbf{k}}, n_3}^* \otimes I_{n_2})), \end{aligned} \quad (21)$$

which implies

$$\begin{aligned}
\text{unfold}(\hat{\mathcal{C}}) &= \text{unfold}(\text{fft}(\mathcal{A} *_\mu \mathcal{B})) \\
&= \sqrt{n_3}(F_{\mu, n_3} \otimes I_{n_1}) \cdot \text{unfold}(\mathcal{A} *_\mu \mathcal{B}) \\
&= \sqrt{n_3}(F_{\mu, n_3} \otimes I_{n_1}) \cdot ((\text{circ}(\mathcal{A}_e) + \text{icirc}(\mathcal{A}_i) \cdot (F_{\mu_i, n_3}^* F_{\mu, n_3} \otimes I_r) \\
&\quad + \text{jcirc}(\mathcal{A}_j) \cdot (F_{\mu_j, n_3}^* F_{\mu, n_3} \otimes I_r) + \text{kcirc}(\mathcal{A}_k) \cdot (F_{\mu_k, n_3}^* F_{\mu, n_3} \otimes I_r)) \cdot \text{unfold}(\mathcal{B})) \\
&= \text{diag}(\hat{\mathcal{A}}) \cdot \sqrt{n_3}(F_{\mu, n_3} \otimes I_r) \cdot \text{unfold}(\mathcal{B}) \\
&= \text{diag}(\hat{\mathcal{A}}) \cdot \text{unfold}(\hat{\mathcal{B}}).
\end{aligned}$$

Thus, the proof is completed. \square

We next to discuss the group-theoretical property of gQt-product in the following theorem. At first, we introduce the concepts of *identity quaternion tensor* and *inverse quaternion tensor*.

Definition 3.2. The $n \times n \times l$ identity quaternion tensor \mathcal{I}_{nnl} is the tensor whose first frontal slice is the identity matrix and others are all zeros.

Definition 3.3. An $n \times n \times l$ quaternion tensor \mathcal{A} is said to be invertible if there exists a quaternion tensor $\mathcal{B} \in \mathbb{H}^{n \times n \times l}$ such that

$$\mathcal{A} *_\mu \mathcal{B} = \mathcal{I}_{nnl} = \mathcal{B} *_mu \mathcal{A}.$$

The tensor \mathcal{B} is called the inverse of \mathcal{A} , denoted as \mathcal{A}^{-1} .

Theorem 3.2. The collection of all invertible $n \times n \times l$ quaternion tensors forms a group under the “ $*_\mu$ ” operation given in Definition 3.1.

Proof. It is easy to see that

$$\text{diag}(\hat{\mathcal{I}}_{nnl}) = I_{nl} \in \mathbb{R}^{nl \times nl}.$$

For all $\mathcal{A} \in \mathbb{H}^{n \times n \times l}$, it follows from Theorem 3.1 that

$$\text{diag}(\hat{\mathcal{A}})\text{diag}(\hat{\mathcal{I}}_{nnl}) = \text{diag}(\hat{\mathcal{I}}_{nnl})\text{diag}(\hat{\mathcal{A}}) = \text{diag}(\hat{\mathcal{A}}) \Leftrightarrow \mathcal{A} *_\mu \mathcal{I}_{nnl} = \mathcal{I}_{nnl} *_mu \mathcal{A} = \mathcal{A},$$

which implies that \mathcal{I}_{nnl} is the identical-element in group.

Similarly, for all $\mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathbb{H}^{n \times n \times l}$, it holds that

$$(\text{diag}(\hat{\mathcal{A}})\text{diag}(\hat{\mathcal{B}}))\text{diag}(\hat{\mathcal{C}}) = \text{diag}(\hat{\mathcal{A}})(\text{diag}(\hat{\mathcal{B}})\text{diag}(\hat{\mathcal{C}})) \Rightarrow (\mathcal{A} *_\mu \mathcal{B}) *_mu \mathcal{C} = \mathcal{A} *_mu (\mathcal{B} *_mu \mathcal{C}).$$

So, the “ $*_\mu$ ” operation is associative. Hence, the collection of all invertible $n \times n \times l$ quaternion tensors forms a group under the “ $*_\mu$ ” operation. \square

We can easily check that the collection of all invertible $n \times n \times l$ quaternion tensors forms a ring under standard tensor addition and gQt-product.

3.3 gQt-SVD: gQt-product based SVD of third-order quaternion tensor

Our goal in this subsection is to build SVD of third-order quaternion tensor based on gQt-product. To begin with, we introduce the concepts of *conjugate transpose* of third-order quaternion tensor and *unitary quaternion tensor*, which will be used in the sequent.

For real third-order tensors, the definition of *conjugate transpose* was given in [10, Definition 3.14], and we recall it in Definition 3.4.

Definition 3.4. Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then $\mathcal{A}^* \in \mathbb{R}^{n_2 \times n_1 \times n_3}$, the conjugate transpose of \mathcal{A} , is obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through n_3 .

Example 3.1. Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times 4}$ and its frontal slices be given by the $n_1 \times n_2$ matrices $A^{(1)}, A^{(2)}, A^{(3)}, A^{(4)}$. Then, the conjugate transpose of \mathcal{A} is

$$\mathcal{A}^* = \text{fold} \left(\begin{bmatrix} (A^{(1)})^* \\ (A^{(4)})^* \\ (A^{(3)})^* \\ (A^{(2)})^* \end{bmatrix} \right).$$

For any $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, it is seen that

$$\text{circ}(\mathcal{A})^* = \text{circ}(\mathcal{A}^*). \quad (22)$$

For any given $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$, by (9), it can be written as $\mathcal{A} = \mathcal{A}_e + \mathcal{A}_i \mathbf{i} + \mathcal{A}_j \mathbf{j} + \mathcal{A}_k \mathbf{k}$ where $\mathcal{A}_e, \mathcal{A}_i, \mathcal{A}_j$ and \mathcal{A}_k are $n_1 \times n_2 \times n_3$ real tensors. Hence, based on Definition 3.4, we can define the conjugate transpose of the quaternion tensor \mathcal{A} .

Definition 3.5. The conjugate transpose of a quaternion tensor $\mathcal{A} = \mathcal{A}_e + \mathcal{A}_i \mathbf{i} + \mathcal{A}_j \mathbf{j} + \mathcal{A}_k \mathbf{k} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ is also denoted as $\mathcal{A}^* \in \mathbb{H}^{n_2 \times n_1 \times n_3}$, which is defined by

$$\text{unfold}(\mathcal{A}^*) = \text{unfold}(\mathcal{A}_e^*) - (T_i^* \otimes I_{n_2})\text{unfold}(\mathcal{A}_i^*)\mathbf{i} - (T_j^* \otimes I_{n_2})\text{unfold}(\mathcal{A}_j^*)\mathbf{j} - (T_k^* \otimes I_{n_2})\text{unfold}(\mathcal{A}_k^*)\mathbf{k},$$

where $T_i^* = F_{\mu, n_3}^* F_{\mu_i, n_3}$, $T_j^* = F_{\mu, n_3}^* F_{\mu_j, n_3}$, and $T_k^* = F_{\mu, n_3}^* F_{\mu_k, n_3}$.

Example 3.2. Let $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times 3}$ and its frontal slices be given by the matrices $A^{(1)}, A^{(2)}, A^{(3)} \in \mathbb{H}^{n_1 \times n_2}$. Setting $\mu = \mathbf{i}$, the conjugate transpose of \mathcal{A} is given as

$$\mathcal{A}^* = \text{fold} \left(\begin{bmatrix} (A_e^{(1)})^* - \mathbf{i}(A_i^{(1)})^* - \mathbf{j}(A_j^{(1)})^* - \mathbf{k}(A_k^{(1)})^* \\ (A_e^{(3)})^* - \mathbf{i}(A_i^{(3)})^* - \mathbf{j}(A_j^{(2)})^* - \mathbf{k}(A_k^{(2)})^* \\ (A_e^{(2)})^* - \mathbf{i}(A_i^{(2)})^* - \mathbf{j}(A_j^{(3)})^* - \mathbf{k}(A_k^{(3)})^* \end{bmatrix} \right).$$

We give some properties of the conjugate transpose of a quaternion tensor \mathcal{A} .

Theorem 3.3. For $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$, we have $\text{diag}(\text{fft}(\mathcal{A}^*)) = \text{diag}(\hat{\mathcal{A}})^*$.

Proof. By (22), (21) and Lemma 2.1, we have

$$\begin{aligned} \text{diag}(\hat{\mathcal{A}})^* &= \left((F_{\mu, n_3} \otimes I_{n_1}) \cdot (\text{circ}(\mathcal{A}_e) \cdot (F_{\mu, n_3}^* \otimes I_{n_2}) + \mathbf{i} \text{circ}(\mathcal{A}_i) \cdot (F_{\mu_i, n_3}^* \otimes I_{n_2}) \right. \\ &\quad \left. + \mathbf{j} \text{circ}(\mathcal{A}_j) \cdot (F_{\mu_j, n_3}^* \otimes I_{n_2}) + \mathbf{k} \text{circ}(\mathcal{A}_k) \cdot (F_{\mu_k, n_3}^* \otimes I_{n_2})) \right)^* \\ &= (F_{\mu, n_3} \otimes I_{n_2}) \cdot \text{circ}(\mathcal{A}_e^*) \cdot (F_{\mu, n_3}^* \otimes I_{n_1}) - (F_{\mu_i, n_3}^* \otimes I_{n_2}) \cdot \text{circ}(\mathcal{A}_i^*) (F_{\mu_i, n_3}^* \otimes I_{n_1}) \mathbf{i} \\ &\quad - (F_{\mu_j, n_3}^* \otimes I_{n_2}) \cdot \text{circ}(\mathcal{A}_j^*) (F_{\mu_j, n_3}^* \otimes I_{n_1}) \mathbf{j} - (F_{\mu_k, n_3}^* \otimes I_{n_2}) \cdot \text{circ}(\mathcal{A}_k^*) (F_{\mu_k, n_3}^* \otimes I_{n_1}) \mathbf{k}. \end{aligned} \quad (23)$$

It follows from (14) and (17) that

$$\begin{aligned} (F_{\mu_i, n_3} \otimes I_{n_2}) \cdot \text{circ}(\mathcal{A}_i^*) \cdot (F_{\mu_i, n_3}^* \otimes I_{n_2}) \mathbf{i} &= \text{diag} \left(\text{fold}(\sqrt{n_3} (F_{\mu_i, n_3} \otimes I_{n_2}) \cdot \text{unfold}(\mathcal{A}_i^*)) \right) \mathbf{i} \\ &= \text{diag} \left(\text{fold}(\sqrt{n_3} (F_{\mu, n_3} F_{\mu_i, n_3}^* \otimes I_{n_2}) \cdot \text{unfold}(\mathcal{A}_i^*) \mathbf{i}) \right) \\ &= \text{diag} \left(\text{fft} \left(\text{fold}((F_{\mu, n_3}^* F_{\mu_i, n_3} \otimes I_{n_2}) \cdot \text{unfold}(\mathcal{A}_i^*) \mathbf{i}) \right) \right). \end{aligned}$$

Similarly,

$$\begin{aligned} (F_{\mu_j, n_3} \otimes I_{n_2}) \cdot \text{circ}(\mathcal{A}_j^*) \cdot (F_{\mu_j, n_3}^* \otimes I_{n_2}) \mathbf{j} &= \text{diag} \left(\text{fft} \left(\text{fold}((F_{\mu, n_3}^* F_{\mu_j, n_3} \otimes I_{n_2}) \cdot \text{unfold}(\mathcal{A}_j^*) \mathbf{j}) \right) \right), \\ (F_{\mu_k, n_3} \otimes I_{n_2}) \cdot \text{circ}(\mathcal{A}_k^*) \cdot (F_{\mu_k, n_3}^* \otimes I_{n_2}) \mathbf{k} &= \text{diag} \left(\text{fft} \left(\text{fold}((F_{\mu, n_3}^* F_{\mu_k, n_3} \otimes I_{n_2}) \cdot \text{unfold}(\mathcal{A}_k^*) \mathbf{k}) \right) \right), \end{aligned}$$

which, together with (23), imply that

$$\text{diag}(\hat{\mathcal{A}})^* = \text{diag}(\text{fft}(\mathcal{A}^*)).$$

Thus, the proof is completed. \square

Corollary 3.1. For any two quaternion tensors \mathcal{A}, \mathcal{B} with adequate dimensions, we have

$$(\mathcal{A} *_\mu \mathcal{B})^* = \mathcal{B}^* *_\mu \mathcal{A}^*.$$

Proof. Since QDFT is invertible, the equality $(\mathcal{A} *_\mu \mathcal{B})^* = \mathcal{B}^* *_\mu \mathcal{A}^*$ can be written as

$$\text{diag}\left(\text{fft}\left((\mathcal{A} *_\mu \mathcal{B})^*\right)\right) = \text{diag}\left(\text{fft}\left(\mathcal{B}^* *_\mu \mathcal{A}^*\right)\right).$$

By Theorems 3.1 and 3.3, we have

$$(\mathcal{A} *_\mu \mathcal{B})^* = \mathcal{B}^* *_\mu \mathcal{A}^* \Leftrightarrow (\text{diag}(\hat{\mathcal{A}})\text{diag}(\hat{\mathcal{B}}))^* = \text{diag}(\hat{\mathcal{B}})^* \text{diag}(\hat{\mathcal{A}})^*.$$

This, together with Lemma 2.1, completes the proof. \square

We next to introduce the concepts of unitary quaternion tensor and partially unitary quaternion tensor. Some properties of unitary quaternion tensor are given, which are still true for partially unitary quaternion tensor.

Definition 3.6. The $n \times n \times l$ quaternion tensor \mathcal{U} is unitary if $\mathcal{U}^* *_\mu \mathcal{U} = \mathcal{U} *_\mu \mathcal{U}^* = \mathcal{I}_{nnl}$. In addition, $\mathcal{U} \in \mathbb{H}^{m \times n \times l}$ is said to be partially unitary if $\mathcal{U}^* *_\mu \mathcal{U} = \mathcal{I}_{nnl}$.

By Theorem 3.3, we immediately obtain the following result.

Corollary 3.2. A quaternion tensor $\mathcal{U} \in \mathbb{H}^{n \times n \times n_3}$ is unitary if and only if $\text{diag}(\hat{\mathcal{U}})$ is a unitary matrix.

A nice feature of unitary quaternion tensor is to preserve the Frobenius norm.

Theorem 3.4. Let \mathcal{U} be a unitary tensor and $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ be a quaternion tensor with adequate dimensions. Then,

$$\|\mathcal{U} *_\mu \mathcal{A}\|_F = \|\mathcal{A}\|_F.$$

Proof. Since normalized QDFT will not change the Frobenius norm, we have

$$\|\mathcal{U} *_\mu \mathcal{A}\|_F^2 = \sum_{p,q} \|(\mathcal{U} *_\mu \mathcal{A})(p, q, :)\|_F^2 = \frac{1}{n_3} \sum_{p,q} \|\text{fft}(\mathcal{U} *_\mu \mathcal{A})(p, q, :)\|_F^2 = \frac{1}{n_3} \|\text{diag}(\text{fft}(\mathcal{U} *_\mu \mathcal{A}))\|_F^2. \quad (24)$$

By Theorems 3.1 and 3.3, Corollary 3.1, and (7), we can obtain

$$\begin{aligned} \|\mathcal{U} *_\mu \mathcal{A}\|_F^2 &= \frac{1}{n_3} \text{tr}\left(\text{diag}(\text{fft}(\mathcal{U} *_\mu \mathcal{A}))^* \cdot \text{diag}(\text{fft}(\mathcal{U} *_\mu \mathcal{A}))\right) \\ &= \frac{1}{n_3} \text{tr}\left(\text{diag}(\text{fft}((\mathcal{U} *_\mu \mathcal{A})^*)) \cdot \text{diag}(\text{fft}(\mathcal{U} *_\mu \mathcal{A}))\right) \\ &= \frac{1}{n_3} \text{tr}\left((\text{diag}(\hat{\mathcal{A}}))^* (\text{diag}(\hat{\mathcal{U}}))^* \text{diag}(\hat{\mathcal{U}}) \text{diag}(\hat{\mathcal{A}})\right) \\ &= \frac{1}{n_3} \|\text{diag}(\hat{\mathcal{A}})\|_F^2 = \|\mathcal{A}\|_F^2. \end{aligned}$$

Thus, we complete the proof. \square

We say a tensor is “f-diagonal” if each frontal slice is diagonal, and then the following decomposition of quaternion tensor is proposed.

Theorem 3.5 (gQt-SVD). Any third-order quaternion tensor $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ can be factorized as

$$\mathcal{A} = \mathcal{U} *_\mu \mathcal{S} *_\mu \mathcal{V}^*,$$

where $\mathcal{S} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor, $\mathcal{U} \in \mathbb{H}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{H}^{n_2 \times n_2 \times n_3}$ are unitary. Moreover, $\|\mathcal{A}\|_F = \|\mathcal{S}\|_F$.

Proof. With the quaternion matrix SVD in Theorem 2.1, we have

$$\text{diag}(\hat{\mathcal{A}}) = \text{diag}(\hat{\mathcal{U}}) \text{diag}(\hat{\mathcal{S}}) \text{diag}(\hat{\mathcal{V}})^*,$$

where $\text{diag}(\hat{\mathcal{U}})$ and $\text{diag}(\hat{\mathcal{V}})^*$ are unitary, and $\text{diag}(\hat{\mathcal{S}})$ is diagonal.

Let $\mathcal{U}, \mathcal{S}, \mathcal{V}$ be the quaternion tensors corresponding to $\text{diag}(\hat{\mathcal{U}})$, $\text{diag}(\hat{\mathcal{S}})$, $\text{diag}(\hat{\mathcal{V}})$, respectively. That is,

$$\begin{aligned}\mathcal{U} &\doteq \text{fold}((F_{\mu, n_3}^* \otimes I_{n_1}) \frac{1}{\sqrt{n_3}} \text{diag}(\hat{\mathcal{U}})(e \otimes I_{n_1})), \\ \mathcal{V} &\doteq \text{fold}((F_{\mu, n_3}^* \otimes I_{n_2}) \frac{1}{\sqrt{n_3}} \text{diag}(\hat{\mathcal{V}})(e \otimes I_{n_2})), \\ \mathcal{S} &\doteq \text{fold}((F_{\mu, n_3}^* \otimes I_{n_1}) \frac{1}{\sqrt{n_3}} \text{diag}(\hat{\mathcal{S}})(e \otimes I_{n_2})),\end{aligned}$$

where e is an n_3 -dimensional column vector whose all elements are 1. Clearly, \mathcal{U}, \mathcal{V} are unitary, and \mathcal{S} is an f-diagonal tensor. It follows from Theorem 3.1 and Corollary 3.2 that $\mathcal{A} = \mathcal{U} *_{\mu} \mathcal{S} *_{\mu} \mathcal{V}^*$. By Theorem 3.4, we immediately obtain $\|\mathcal{A}\|_F = \|\mathcal{S}\|_F$. \square

By Theorem 3.5, any third-order quaternion tensor has gQt-SVD. So, in the next subsection we will define its rank based on such decomposition, and show the existence of low-rank optimal approximation.

3.4 gQt-rank, low-rank optimal approximation, and multi-gQt-rank

For any third-order quaternion tensor $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ and its gQt-SVD $\mathcal{A} = \mathcal{U} *_{\mu} \mathcal{S} *_{\mu} \mathcal{V}^*$, we regard $\mathcal{U}(:, i, :) \in \mathbb{H}^{n_1 \times 1 \times n_3}$, $\mathcal{V}(:, i, :) \in \mathbb{H}^{n_2 \times 1 \times n_3}$, and $\mathcal{S}(i, i, :) \in \mathbb{H}^{1 \times 1 \times n_3}$ as tensors. By QDFT (11), we have

$$\text{fft}(\mathcal{U}(:, i, :)) = \hat{\mathcal{U}}(:, i, :), \quad \text{fft}(\mathcal{S}(i, i, :)) = \hat{\mathcal{S}}(i, i, :), \quad \text{fft}(\mathcal{V}(:, i, :)) = \hat{\mathcal{V}}(:, i, :).$$

It follows from Theorem 2.1 that

$$\text{diag}(\hat{\mathcal{A}}) = \sum_{i=1}^{\min(n_1, n_2)} \text{diag}(\text{fft}(\mathcal{U}(:, i, :))) \text{diag}(\text{fft}(\mathcal{S}(i, i, :))) \text{diag}(\text{fft}(\mathcal{V}(:, i, :)))^*.$$

With simple calculation, we can get

$$\mathcal{A} = \sum_{i=1}^m \mathcal{U}(:, i, :) *_{\mu} \mathcal{S}(i, i, :) *_{\mu} \mathcal{V}(:, i, :)^*, \quad m \doteq \min(n_1, n_2). \quad (25)$$

Thus, \mathcal{A} can be written as a finite sum of gQt-product of matrices. Naturally, according to (25), we can define the “gQt-rank” for any third-order quaternion tensor.

Definition 3.7 (gQt-rank). Let $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ and its gQt-SVD be given as $\mathcal{A} = \mathcal{U} *_{\mu} \mathcal{S} *_{\mu} \mathcal{V}^*$. The number of nonzero elements of $\{\mathcal{S}(i, i, :)\}_{i=1}^m$ is called gQt-rank of \mathcal{A} , denote as $\text{rank}_{gQt}(\mathcal{A})$. That is,

$$\text{rank}_{gQt}(\mathcal{A}) \doteq \#\{i \mid \mathcal{S}(i, i, :) \neq 0\} = \#\{i \mid \hat{\mathcal{S}}(i, i, :) \neq 0\}.$$

The i -th singular value of \mathcal{A} is defined as

$$\sigma_i(\mathcal{A}) \doteq \frac{1}{n_3} \|\hat{\mathcal{S}}(i, i, :)\|_1, \quad i \in [m],$$

and the nuclear norm of \mathcal{A} is defined as

$$\|\mathcal{A}\|_* \doteq \sum_{i=1}^m \sigma_i(\mathcal{A}).$$

Similar to Lemma 2.2, we have the following result for the nuclear norm $\|\mathcal{A}\|_*$.

Lemma 3.2. Let $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$, then

$$\|\mathcal{A}\|_* = \frac{1}{n_3} \sum_{i=1}^{n_3} \|\hat{\mathcal{A}}(:, :, i)\|_* = \min_{\mathcal{X}, \mathcal{Y}} \left\{ \frac{1}{2} (\|\mathcal{X}\|_F^2 + \|\mathcal{Y}\|_F^2) : \mathcal{A} = \mathcal{X} *_{\mu} \mathcal{Y}^* \right\}.$$

Proof. From Definition 3.7, we have

$$\begin{aligned}
\|\mathcal{A}\|_* &= \sum_{j=1}^m \sum_{i=1}^{n_3} |\hat{\mathcal{S}}(j, i)| = \frac{1}{n_3} \sum_{i=1}^{n_3} \|\hat{\mathcal{A}}(:, :, i)\|_* \\
&= \min_{\hat{\mathcal{X}}, \hat{\mathcal{Y}}} \left\{ \frac{1}{2n_3} \sum_{i=1}^{n_3} (\|\hat{\mathcal{X}}(:, :, i)\|_F^2 + \|\hat{\mathcal{Y}}(:, :, i)\|_F^2) : \hat{\mathcal{X}}(:, :, i)(\hat{\mathcal{Y}}(:, :, i))^* = \hat{\mathcal{A}}(:, :, i), i \in [n_3] \right\} \\
&= \min_{\mathcal{X}, \mathcal{Y}} \left\{ \frac{1}{2} (\|\mathcal{X}\|_F^2 + \|\mathcal{Y}\|_F^2) : \mathcal{A} = \mathcal{X} *_\mu \mathcal{Y}^* \right\}.
\end{aligned}$$

□

By (25), the following theorem shows the existence of low-rank ($\text{rank}_{gQt}(\mathcal{A}) = k < m$) optimal approximation.

Theorem 3.6. Let $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ and its gQt -SVD be given as $\mathcal{A} = \mathcal{U} *_\mu \mathcal{S} *_\mu \mathcal{V}^*$. For $k < m$, denote

$$\mathcal{A}_k = \sum_{i=1}^k \mathcal{U}(:, i, :)_\mu \mathcal{S}(i, i, :)_\mu \mathcal{V}(:, i, :)^*. \quad (26)$$

Then,

$$\mathcal{A}_k \in \arg \min_{\mathcal{C} \in M_k} \|\mathcal{A} - \mathcal{C}\|_F, \quad \text{where } M_k = \{\mathcal{C} | \mathcal{X} \in \mathbb{H}^{n_1 \times k \times n_3}, \mathcal{Y} \in \mathbb{H}^{k \times n_2 \times n_3}, \mathcal{C} = \mathcal{X} *_\mu \mathcal{Y}\}.$$

Proof. For all $\mathcal{C} \in M_k$, it is easy to see that the rank of each block matrix of $\text{diag}(\hat{\mathcal{C}})$ is not greater than k . Since there exists rank- k optimal approximation of quaternion matrix, it follows that

$$\begin{aligned}
\|\mathcal{A} - \mathcal{C}\|_F^2 &= \frac{1}{n_3} \|\text{diag}(\hat{\mathcal{A}}) - \text{diag}(\hat{\mathcal{C}})\|_F^2 \\
&= \frac{1}{n_3} \sum_{q=1}^{n_3} \|\hat{\mathcal{A}}(:, :, q) - \hat{\mathcal{C}}(:, :, q)\|_F^2 \\
&\geq \frac{1}{n_3} \sum_{q=1}^{n_3} \|\hat{\mathcal{A}}(:, :, q) - \sum_{p=1}^k \hat{\mathcal{U}}(:, p, q) \hat{\mathcal{S}}(p, p, q) \hat{\mathcal{V}}(:, p, q)^*\|_F^2 \\
&= \frac{1}{n_3} \sum_{q=1}^{n_3} \|\hat{\mathcal{A}}(:, :, q) - \hat{\mathcal{A}}_k(:, :, q)\|_F^2 = \|\mathcal{A} - \mathcal{A}_k\|_F^2,
\end{aligned}$$

where the first equality is from (24) and the last two equalities hold due to (26) and Theorem 3.1. Thus, this completes the proof. □

We now investigate gQt -rank of third order quaternion tensor generated by color video, and show that the low gQt -rankness is actually an inherent property of many color videos. We select a video data “News” in widely used YUV Video Sequences¹. We take all 300 frames of size 288×352 as a color video data $\tilde{\mathcal{C}}$, i.e., $\tilde{\mathcal{C}} \in \mathbb{H}^{288 \times 352 \times 300}$, where we encode the red, green, and blue channel pixel values on the three imaginary parts of a quaternion. Figure 1 illustrates the low-rank structure of $\tilde{\mathcal{C}}$ from color videos.

¹<http://trace.eas.asu.edu/yuv/>

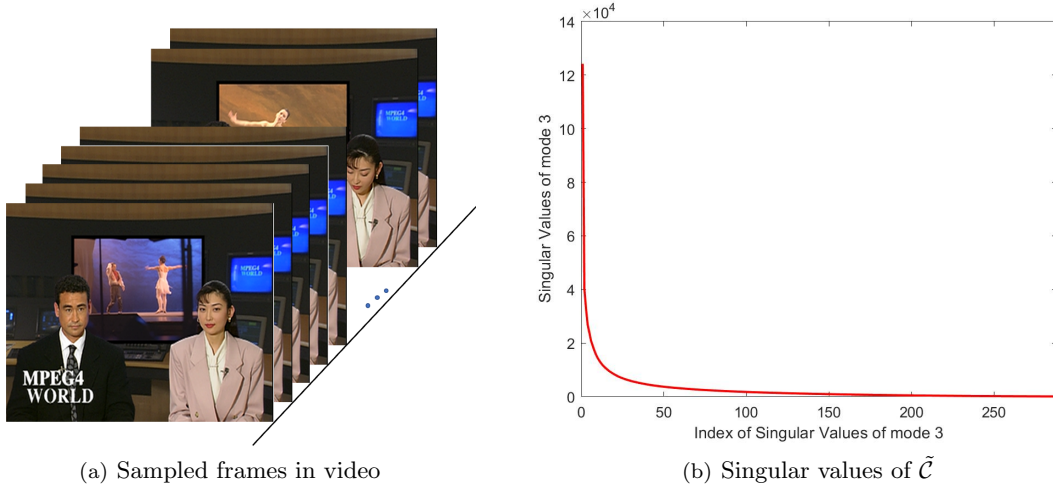


Figure 1: The sampled frames in video and singular values.

Note that gQt-rank is only defined on one mode-3 of third order quaternion tensor, and the low-rank structure on the other two modes is missing. Motivated by this and *multi-tubal rank* given [33], we next to introduce “multi-gQt-rank” for third-order quaternion tensor $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ by extending QDFT (11) from mode-3 to the other two modes.

To begin with some notations, we denote

$$A_1^{(i)} = \mathcal{A}(i, :, :), \quad A_2^{(j)} = \mathcal{A}(:, j, :), \quad A_3^{(k)} = \mathcal{A}(:, :, k), \quad i \in [n_1], j \in [n_2], k \in [n_3].$$

Define QDFT of \mathcal{A} along w -th mode ($w = 1, 2, 3$) as $\hat{\mathcal{A}}_1$, $\hat{\mathcal{A}}_2$ and $\hat{\mathcal{A}}_3$, which satisfy

$$\hat{\mathcal{A}}_1(:, j, k) = F_{\mu, n_1} \mathcal{A}(:, j, k), \quad \hat{\mathcal{A}}_2(i, :, k) = F_{\mu, n_2} \mathcal{A}(i, :, k), \quad \hat{\mathcal{A}}_3(i, j, :) = F_{\mu, n_3} \mathcal{A}(i, j, :),$$

for $i \in [n_1]$, $j \in [n_2]$, $k \in [n_3]$. Here, F_{μ, n_w} is defined similarly to (11). For simplicity, we define

$$\hat{\mathcal{A}}_1^{(i)} \doteq \hat{\mathcal{A}}_1(i, :, :), \quad \hat{\mathcal{A}}_2^{(j)} \doteq \hat{\mathcal{A}}_2(:, j, :), \quad \hat{\mathcal{A}}_3^{(k)} \doteq \hat{\mathcal{A}}_3(:, :, k), \quad i \in [n_1], j \in [n_2], k \in [n_3].$$

We also define the following operators for $w = 1, 2, 3$,

$$\text{diag}_w(\mathcal{A}) \doteq \begin{bmatrix} A_w^{(1)} & & & \\ & A_w^{(2)} & & \\ & & \ddots & \\ & & & A_w^{(n_w)} \end{bmatrix}, \quad \text{circ}_w(\mathcal{A}) \doteq \begin{bmatrix} A_w^{(1)} & A_w^{(n_3)} & \cdots & A_w^{(2)} \\ A_w^{(2)} & A_w^{(1)} & \cdots & A_w^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ A_w^{(n_3)} & A_w^{(n_3-1)} & \cdots & A_w^{(1)} \end{bmatrix},$$

and

$$\text{unfold}_w(\mathcal{A}) \doteq [A_w^{(1)}; A_w^{(2)}; \dots; A_w^{(n_w)}].$$

The inverse operator “fold_w” is defined by $\text{fold}_w(\text{unfold}_w(\mathcal{A})) = \mathcal{A}$. With μ_i, μ_j, μ_k defined in (15), we set

$$T_{i,w} = F_{\mu_i, n_w}^* F_{\mu, n_w}, \quad T_{j,w} = F_{\mu_j, n_w}^* F_{\mu, n_w}, \quad T_{k,w} = F_{\mu_k, n_w}^* F_{\mu, n_w}.$$

We now generalize gQt-product along three modes. For $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times r}$ and $\mathcal{B} \in \mathbb{H}^{n_1 \times r \times n_3}$, define

$$\mathcal{A} *_\mu^1 \mathcal{B} \doteq \text{fold}((\text{circ}_1(\mathcal{A}_e) + i \text{circ}_1(\mathcal{A}_i) \cdot (T_{i,1} \otimes I_r) + j \text{circ}_1(\mathcal{A}_j) \cdot (T_{j,1} \otimes I_r) + k \text{circ}_1(\mathcal{A}_k) \cdot (T_{k,1} \otimes I_r)) \cdot \text{unfold}_1(\mathcal{B})).$$

For $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times r}$ and $\mathcal{B} \in \mathbb{H}^{r \times n_2 \times n_3}$, define

$$\mathcal{A} *_\mu^2 \mathcal{B} \doteq \text{fold}((\text{circ}_2(\mathcal{A}_e) + i \text{circ}_2(\mathcal{A}_i) \cdot (T_{i,2} \otimes I_r) + j \text{circ}_2(\mathcal{A}_j) \cdot (T_{j,2} \otimes I_r) + k \text{circ}_2(\mathcal{A}_k) \cdot (T_{k,2} \otimes I_r)) \cdot \text{unfold}_2(\mathcal{B})).$$

For $\mathcal{A} \in \mathbb{H}^{n_1 \times r \times n_3}$ and $\mathcal{B} \in \mathbb{H}^{r \times n_2 \times n_3}$, define

$$\mathcal{A} *_\mu^3 \mathcal{B} \doteq \text{fold}((\text{circ}_3(\mathcal{A}_e) + i \text{circ}_3(\mathcal{A}_i) \cdot (T_{i,3} \otimes I_r) + j \text{circ}_3(\mathcal{A}_j) \cdot (T_{j,3} \otimes I_r) + k \text{circ}_3(\mathcal{A}_k) \cdot (T_{k,3} \otimes I_r)) \cdot \text{unfold}_3(\mathcal{B})).$$

We introduce the concept of “multi-gQt-rank” for third-order quaternion tensor as follows.

Definition 3.8 (multi-gQt-rank). Let $\mathcal{A} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ and $r_w^l = \text{rank}(\hat{A}_w^{(l)})$ with $l \in [n_w]$ and $w \in [3]$. The multi-gQt-rank of \mathcal{A} is defined as

$$\text{rank}_{mgQt}(\mathcal{A}) = (r_1(\mathcal{A}), r_2(\mathcal{A}), r_3(\mathcal{A})),$$

where $r_w(\mathcal{A}) = \max(r_w^1, r_w^2, \dots, r_w^{n_w})$.

By Theorem 3.1 and Lemma 3.2, we can easily get the following results.

Theorem 3.7. Let \mathcal{A}, \mathcal{B} be quaternion tensors and $\mathcal{C} = \mathcal{A} *_{\mu}^w \mathcal{B}$ be defined above for $w \in [3]$. If their QDFT tensors along mode- w are $\hat{\mathcal{A}}_w, \hat{\mathcal{B}}_w$, and $\hat{\mathcal{C}}_w$, respectively, then, it holds

- (i) $\|\mathcal{A}\|_F^2 = \frac{1}{n_w} \|\hat{\mathcal{A}}_w\|_F^2$.
- (ii) $\text{diag}_w(\hat{\mathcal{C}}_w) = \text{diag}_w(\hat{\mathcal{A}}_w) \cdot \text{diag}_w(\hat{\mathcal{B}}_w)$.
- (iii) $r_w(\mathcal{C}) \leq \min(r_w(\mathcal{A}), r_w(\mathcal{B}))$.

We now investigate multi-gQt-rank of third order quaternion tensor generated by color video, and show that the low rankness is actually an inherent property of many color videos. We select a video data “Coastguard” in YUV Video Sequences² to group into the quaternion tensor $\tilde{\mathcal{C}} \in \mathbb{H}^{288 \times 352 \times 300}$. Figure 2 shows the low-rank structures of tensor $\tilde{\mathcal{C}}$ in mode 1, 2 and 3.

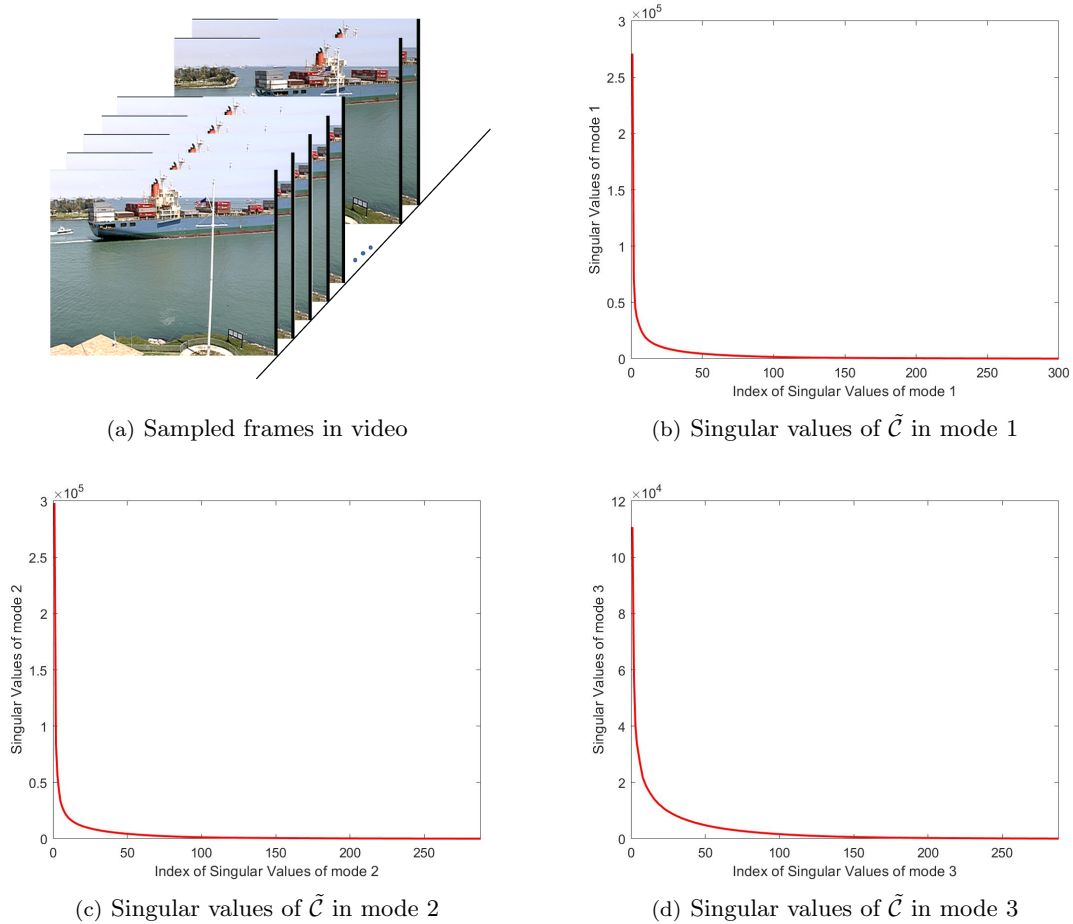


Figure 2: The sampled frames in video and singular values.

As mentioned in [3, 34, 35], when the color image or video data is regarded as quaternion matrices or real tensors, they lie on a union of low-rank subspaces approximately, which lead the low-rank structure

²<http://trace.eas.asu.edu/yuv/>

of the real data. This is also true for third-order quaternion tensors data. Figures 1 and 2 indicate that third-order quaternion tensors generated by color videos in real life have an inherent low-rank property. Hence, in order to recover color videos with partial data loss, we can design tensor completion model as (2) via gQt-rqnk and multi-gQt-rank. This is the aim of next section.

4 Quaternion Tensor Completion for Color Video Inpainting

In this section, we establish low-rank quaternion tensor completion models based on gQt-rank and multi-gQt-rank to recover color videos with partial data loss. We present an ALS algorithm to solve the proposed models and show that the generated sequence converges to the stationary point of our model.

4.1 Low gQt-rank quaternion tensor completion model

We first define the operator \Re to get the real part of a quaternion tensor, and the operator \Im to get the imaginary part. Our low gQt-rank quaternion tensor completion is to find the minimal gQt-rank solution satisfying the consistency with the observed data. Let \mathcal{M} be the raw tensor and Ω be the index set locating the observed data. Then, by (2), the low gQt-rank quaternion tensor completion can be modeled as

$$\min_{\mathcal{C} \in \mathbb{H}^{n_1 \times n_2 \times n_3}} \text{rank}_{gQt}(\mathcal{C}), \quad \text{s.t. } P_{\Omega}(\mathcal{C} - \mathcal{M}) = 0, \Re(\mathcal{C}) = 0, \quad (27)$$

To solve (27) efficiently, the following relaxation via nuclear norm is considered

$$\min_{\mathcal{C} \in \mathbb{H}^{n_1 \times n_2 \times n_3}} \|\mathcal{C}\|_*, \quad \text{s.t. } P_{\Omega}(\mathcal{C} - \mathcal{M}) = 0, \Re(\mathcal{C}) = 0.$$

By Theorem 3.5, computing $\|\mathcal{C}\|_*$ is via gQt-SVD of \mathcal{C} . However, its computational complexity will be $O(n_1 n_2 n_3 (\log(n_3) + \min(n_1, n_2)))$. In order to reduce the computational cost, by Lemma 3.2, we consider the following quaternion tensor factorization model:

$$\min_{\mathcal{C}, \mathcal{A}, \mathcal{B}} \frac{1}{2} (\|\mathcal{A}\|_F^2 + \|\mathcal{B}\|_F^2), \quad \text{s.t. } \mathcal{C} = \mathcal{A} *_{\mu} \mathcal{B}, P_{\Omega}(\mathcal{C} - \mathcal{M}) = 0, \Re(\mathcal{C}) = 0.$$

In practical applications, each frame of the video data has spatial stability feature. We use total variation (TV) to capture these spatial correlation features and consider the square of total variation of data to keep objective function analytic, i.e.,

$$\|\mathcal{C} \times_1 H_{n_1}\|_F^2 + \|\mathcal{C} \times_2 H_{n_2}\|_F^2 = \sum_{k=1}^{n_1-1} \|\mathcal{C}(k, :, :) - \mathcal{C}(k+1, :, :)\|_F^2 + \sum_{k=1}^{n_2-1} \|\mathcal{C}(:, k, :) - \mathcal{C}(:, k+1, :)\|_F^2,$$

where $H = \text{Toeplitz}(0, -1, 1)$ be an $(n-1) \times n$ Toeplitz matrix, i.e.,

$$H_n = \begin{bmatrix} 1 & -1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & -1 & \ddots & \ddots & 0 \\ 0 & 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & \cdots & \cdots & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(n-1) \times n}.$$

In addition, the real color video quaternion tensor \mathcal{C} is only approximately low gQt-rank, and hence it is likely to fail to find a low gQt-rank solution strictly satisfying the restriction $\mathcal{C} = \mathcal{A} *_{\mu} \mathcal{B}$. Therefore, we usually penalize $\mathcal{C} = \mathcal{A} *_{\mu} \mathcal{B}$ into objective function. Thus, we get our final low gQt-rank quaternion tensor completion model as follows:

$$\begin{aligned} \min_{\mathcal{C}, \mathcal{A}, \mathcal{B}} \quad & f(\mathcal{C}, \mathcal{A}, \mathcal{B}) \doteq \frac{1}{2} \|\mathcal{A} *_{\mu} \mathcal{B} - \mathcal{C}\|_F^2 + \frac{\lambda}{2} (\|\mathcal{A}\|_F^2 + \|\mathcal{B}\|_F^2) + \sum_{k=1}^2 \lambda_k \|\mathcal{C} \times_k H_{n_k}\|_F^2, \\ \text{s.t.} \quad & P_{\Omega}(\mathcal{C} - \mathcal{M}) = 0, \Re(\mathcal{C}) = 0, \end{aligned} \quad (28)$$

where λ , λ_1 and λ_2 are the penalty parameters.

We will propose an algorithm for solving the model (28) in the next subsection.

4.2 Solution method

In this subsection, we present an ALS procedure to solve (28). At each iteration, two variables of \mathcal{A} , \mathcal{B} , \mathcal{C} are fixed and the other one is updated by solving the updated model (28).

At the t -th iteration of our method, \mathcal{C}^t is updated by

$$\mathcal{C}^t = \arg \min_{P_\Omega(\mathcal{C}-\mathcal{M})=0, \Re(\mathcal{C})=0} \frac{1}{2} \|\mathcal{A}^{t-1} *_{\mu} \mathcal{B}^{t-1} - \mathcal{C}\|_F^2 + \sum_{k=1}^2 \lambda_k \|\mathcal{C} \times_k H_{n_k}\|_F^2, \quad (29)$$

and $\mathcal{A}^t, \mathcal{B}^t$ are updated by the regularized version of (28) as follows:

$$\mathcal{A}^t = \arg \min \frac{1}{2} \|\mathcal{A} *_{\mu} \mathcal{B}^{t-1} - \mathcal{C}^t\|_F^2 + \frac{\lambda}{2} \|\mathcal{A}\|_F^2 + \frac{\beta}{2} \|\mathcal{A} - \mathcal{A}^{t-1}\|_F^2, \quad (30)$$

$$\mathcal{B}^t = \arg \min \frac{1}{2} \|\mathcal{A}^{t-1} *_{\mu} \mathcal{B} - \mathcal{C}^t\|_F^2 + \frac{\lambda}{2} \|\mathcal{B}\|_F^2 + \frac{\beta}{2} \|\mathcal{B} - \mathcal{B}^{t-1}\|_F^2, \quad (31)$$

where $\beta > 0$ is the regularization parameter.

We next to solve the subproblems (29)-(31). First, we rewrite (29) as

$$\mathcal{C}^t = \arg \min_{\Re(\mathcal{C})=0} \frac{1}{2} \|\mathcal{A}^{t-1} *_{\mu} \mathcal{B}^{t-1} - \mathcal{C}\|_F^2 + \sum_{k=1}^2 \lambda_k \|\mathcal{C} \times_k H_{n_k}\|_F^2 + \delta_{\{P_\Omega(\mathcal{C}-\mathcal{M})=0\}}(\mathcal{C}), \quad (32)$$

where $\delta_S(\cdot)$ is the indicator function, i.e., $\delta_S(x) = 0$, if $x \in S$; $\delta_S(x) = +\infty$, if $x \notin S$.

For simplicity, set $\mathcal{C} = \mathcal{C}_i \mathbf{i} + \mathcal{C}_j \mathbf{j} + \mathcal{C}_k \mathbf{k}$ whose real part is zero. Thus, the objective function in (28) can be written as $f(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k | \hat{\mathcal{A}}, \hat{\mathcal{B}})$ due to $\sqrt{n_3} \|\mathcal{A}\|_F = \|\hat{\mathcal{A}}\|_F$. Denote

$$\begin{aligned} f_1(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k) &= \frac{1}{2} \|\Im(\mathcal{A}^{t-1} *_{\mu} \mathcal{B}^{t-1}) - \mathcal{C}\|_F^2 + \sum_{k=1}^2 \lambda_k \|\mathcal{C} \times_k H_{n_k}\|_F^2, \\ h_1(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k) &= \delta_{\{P_\Omega(\mathcal{C}-\mathcal{M})=0\}}(\mathcal{C}). \end{aligned} \quad (33)$$

Then, (32) is equivalent to the following unconstrained optimization problem:

$$\min_{\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k} f_1(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k) + h_1(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k). \quad (34)$$

Clearly, f_1 is differentiable and h_1 is a proper closed convex function. So, we can solve (34) inexactly by the well-known proximal gradient method (PGM). We employ the following PGM with Barzilar-Borwein [36] line research rule to solve (34).

Algorithm 1 BB-PGM for (34)

Input. The tensor data $\mathcal{M}, \mathcal{A}^{t-1} *_{\mu} \mathcal{B}^{t-1}$, the observed set Ω , initial step size α_0 , parameters $\lambda, \lambda_i, H_{n_i}, 1 \leq i \leq 2$.

Step 0. Initialize $z^0 = [\mathcal{C}_i^0, \mathcal{C}_j^0, \mathcal{C}_k^0]$ which satisfies $P_\Omega(\mathcal{C}_i^0 \mathbf{i} + \mathcal{C}_j^0 \mathbf{j} + \mathcal{C}_k^0 \mathbf{k} - \mathcal{M}) = 0$, $z^{-1} = z^0$. Iterate the following steps for $k = 0, 1, 2, \dots$ while it does not satisfy stop criterion.

Step 1. If $k \geq 1$, choose step size

$$\alpha_k = \frac{\|z^k - z^{k-1}\|_F^2}{\langle z^k - z^{k-1}, \nabla f_1(z^k) - \nabla f_1(z^{k-1}) \rangle}, \text{ or } \alpha_k = \frac{\langle z^k - z^{k-1}, \nabla f_1(z^k) - \nabla f_1(z^{k-1}) \rangle}{\|\nabla f_1(z^k) - \nabla f_1(z^{k-1})\|_F^2}.$$

Step 2. Update $[\mathcal{C}_i^{k+1}, \mathcal{C}_j^{k+1}, \mathcal{C}_k^{k+1}] = \text{prox}_{\alpha_k h_1}([\mathcal{C}_i^k, \mathcal{C}_j^k, \mathcal{C}_k^k] - \alpha_k \nabla f_1(\mathcal{C}_i^k, \mathcal{C}_j^k, \mathcal{C}_k^k))$.

Output. $\mathcal{C}_i^{k+1} \mathbf{i} + \mathcal{C}_j^{k+1} \mathbf{j} + \mathcal{C}_k^{k+1} \mathbf{k}$.

Here, $\text{prox}_{\alpha_k h_1}(\cdot)$ in Step 2 of Algorithm 1 is the proximal mapping of h_1 with parameter $\alpha_k > 0$, i.e.,

$$\text{prox}_{\alpha_k h_1}(\mathcal{X}) = \arg \min_{\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times 3}} \left\{ \alpha_k h_1(\mathcal{Y}) + \frac{1}{2} \|\mathcal{Y} - \mathcal{X}\|_F^2 \right\}, \quad \forall \mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times 3}.$$

It is known that the proximal mapping of indicator function is a projection, i.e.,

$$\text{prox}_{\alpha_k h_1}(\mathcal{X}) = P_{\{P_\Omega(\mathcal{X}_i \mathbf{i} + \mathcal{X}_j \mathbf{j} + \mathcal{X}_k \mathbf{k} - \mathcal{M}) = 0\}}(\mathcal{X}), \quad \forall \mathcal{X} = [\mathcal{X}_i, \mathcal{X}_j, \mathcal{X}_k] \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times 3}.$$

For solving (30) and (31), we consider their matrix versions because the updates of \mathcal{A}^t and \mathcal{B}^t are just those of their QDFT tensors from the calculation perspective. Hence, by (13), we can rewrite (30) and (31) as the following corresponding matrix versions:

$$\hat{\mathcal{A}}^t = \arg \min_{\hat{\mathcal{A}}} \sum_{l=1}^{n_3} \left(\frac{1}{2} \|\hat{\mathcal{A}}^{(l)} \hat{\mathcal{B}}^{t-1, (l)} - \hat{\mathcal{C}}^{t, (l)}\|_F^2 + \frac{\lambda}{2} \|\hat{\mathcal{A}}^{(l)}\|_F^2 + \frac{\beta}{2} \|\hat{\mathcal{A}}^{(l)} - \hat{\mathcal{A}}^{t-1, (l)}\|_F^2 \right), \quad (35)$$

and

$$\hat{\mathcal{B}}^t = \arg \min_{\hat{\mathcal{B}}} \sum_{l=1}^{n_3} \left(\frac{1}{2} \|\hat{\mathcal{A}}^{t, (l)} \hat{\mathcal{B}}^{(l)} - \hat{\mathcal{C}}^{t, (l)}\|_F^2 + \frac{\lambda}{2} \|\hat{\mathcal{B}}^{(l)}\|_F^2 + \frac{\beta}{2} \|\hat{\mathcal{B}}^{(l)} - \hat{\mathcal{B}}^{t-1, (l)}\|_F^2 \right). \quad (36)$$

To solve the problems (35) and (36) with quaternion variables, we apply the following results, which were given in [37] to introduce the gradient for a quaternion matrix function and optimality condition for an equality-constrained quaternion matrix optimization.

Definition 4.1 ([37] Definition 4.1). *Let $f : \mathbb{H}^{m \times n} \rightarrow \mathbb{R}$ and $X = X_e + X_i \mathbf{i} + X_j \mathbf{j} + X_k \mathbf{k}$. f is said to be differentiable at X if $\frac{\partial f}{\partial X_v}$ exists at X_v for $v = \mathbf{e}, \mathbf{i}, \mathbf{j}, \mathbf{k}$. Moreover, its gradient is defined as*

$$\nabla_{\mathbb{H}} f(X) = \frac{\partial f}{\partial X_e} + \frac{\partial f}{\partial X_i} \mathbf{i} + \frac{\partial f}{\partial X_j} \mathbf{j} + \frac{\partial f}{\partial X_k} \mathbf{k}.$$

f is said to be continuously differentiable at X if $\frac{\partial f}{\partial X_v}$ exists in a neighborhood of X_v and is continuous at X_v for $v = \mathbf{e}, \mathbf{i}, \mathbf{j}, \mathbf{k}$. Furthermore, f is said to be continuously differentiable if f is continuously differentiable at any $X \in \mathbb{H}^{m \times n}$.

Theorem 4.1 ([37] Theorem 4.2). *Suppose that $f : \mathbb{H}^{m \times n} \rightarrow \mathbb{R}$ is continuously differentiable, and $X^\# \in \mathbb{H}^{m \times n}$ is an optimal solution of $\min\{f(X)\}$. Then, it holds*

$$\nabla_{\mathbb{H}} f(X^\#) = O.$$

By Theorem 4.1, we can find the closed-form solutions to (30) and (31). For $l \in [n_3]$, $\hat{\mathcal{A}}^{t, (l)}$ is updated as

$$\begin{aligned} \hat{\mathcal{A}}^{t, (l)} &= \arg \min_{\hat{\mathcal{A}}} \frac{1}{2} \|\hat{\mathcal{A}}^{(l)} \hat{\mathcal{B}}^{t-1, (l)} - \hat{\mathcal{C}}^{t, (l)}\|_F^2 + \frac{\lambda}{2} \|\hat{\mathcal{A}}^{(l)}\|_F^2 + \frac{\beta}{2} \|\hat{\mathcal{A}}^{(l)} - \hat{\mathcal{A}}^{t-1, (l)}\|_F^2 \\ &= (\hat{\mathcal{C}}^{t, (l)} (\hat{\mathcal{B}}^{t-1, (l)})^* + \beta \hat{\mathcal{A}}^{t-1, (l)}) (\hat{\mathcal{B}}^{t-1, (l)} (\hat{\mathcal{B}}^{t-1, (l)})^* + (\lambda + \beta) I)^{-1}, \end{aligned} \quad (37)$$

and $\hat{\mathcal{B}}^{t, (l)}$ is updated as

$$\begin{aligned} \hat{\mathcal{B}}^{t, (l)} &= \arg \min_{\hat{\mathcal{B}}} \frac{1}{2} \|\hat{\mathcal{A}}^{t, (l)} \hat{\mathcal{B}}^{(l)} - \hat{\mathcal{C}}^{t, (l)}\|_F^2 + \frac{\lambda}{2} \|\hat{\mathcal{B}}^{(l)}\|_F^2 + \frac{\beta}{2} \|\hat{\mathcal{B}}^{(l)} - \hat{\mathcal{B}}^{t-1, (l)}\|_F^2 \\ &= ((\hat{\mathcal{A}}^{t, (l)})^* \hat{\mathcal{A}}^{t, (l)} + (\lambda + \beta) I)^{-1} ((\hat{\mathcal{A}}^{t, (l)})^* \hat{\mathcal{C}}^{t, (l)} + \beta \hat{\mathcal{B}}^{t-1, (l)}). \end{aligned} \quad (38)$$

Denote Ω^c as the complement of the set Ω . Based on above discussions, we propose the following algorithm to solve our model (28).

4.3 Convergence analysis for QRTC

We now analyze the convergence of QRTC. For convenience, we collect all variables as a real undetermined vector

$$z \doteq (C_i, C_j, C_k, \hat{\mathcal{A}}_e, \hat{\mathcal{A}}_i, \hat{\mathcal{A}}_j, \hat{\mathcal{A}}_k, \hat{\mathcal{B}}_e, \hat{\mathcal{B}}_i, \hat{\mathcal{B}}_j, \hat{\mathcal{B}}_k) \in \mathbb{R}^{3n_1 n_2 n_3 + 4\|\mathbf{r}\|_1(n_1 + n_2)}.$$

Algorithm 2 gQt-Rank Tensor Completion (QRTC)

Input. The tensor data $\mathcal{M} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$, the observed set Ω , the rank $\mathbf{r} \in \mathbb{Z}_+^{n_3}$, parameters $\lambda, \lambda_i, H_{n_i}, 1 \leq i \leq 3$ and ϵ .

Step 0. Initialize $\hat{\mathcal{A}}^0, \hat{\mathcal{B}}^0$ and \mathcal{C}^0 satisfying $P_\Omega(\mathcal{C}^0 - \mathcal{M}) = 0$, $\Re(\mathcal{C}^0) = 0$ and the rank of $\hat{\mathcal{A}}^0, \hat{\mathcal{B}}^0$ are less than \mathbf{r} . Iterate the following steps for $t = 1, 2, \dots$ while it does not satisfy stop criterion.

Step 1. Compute

$$\mathcal{C}^t = \arg \min_{P_\Omega(\mathcal{C} - \mathcal{M})=0, \Re(\mathcal{C})=0} \frac{1}{2} \|\mathcal{A}^{t-1} *_\mu \mathcal{B}^{t-1} - \mathcal{C}\|_F^2 + \sum_{k=1}^2 \lambda_k \|\mathcal{C} \times_k H_{n_k}\|_F^2 + \sum_{\alpha=\mathbf{i}, \mathbf{j}, \mathbf{k}} \langle \delta_\alpha^t, \mathcal{C}_\alpha \rangle \quad (39)$$

via Algorithm 1. That is, given parameters of Algorithm 1, apply BB-PGM to find an approximate solution \mathcal{C}^t of (32) such that the error vector δ^t satisfies $\Re(\delta^t) = 0$, $P_\Omega(\delta^t) = 0$ and the accuracy condition

$$\|P_{\Omega^c}(\delta^t)\|_F \leq \frac{1}{4} \|\mathcal{C}^t - \mathcal{C}^{t-1}\|_F. \quad (40)$$

Step 2. Compute $\hat{\mathcal{A}}^t$ by (37).

Step 3. Compute $\hat{\mathcal{B}}^t$ by (38).

Step 5. Check the stop criterion: $|f(\mathcal{C}^k, \mathcal{A}^k, \mathcal{B}^k) - f(\mathcal{C}^{k-1}, \mathcal{A}^{k-1}, \mathcal{B}^{k-1})| / |f(\mathcal{C}^{k-1}, \mathcal{A}^{k-1}, \mathcal{B}^{k-1})| < \epsilon$.

Output. \mathcal{C}^t .

And then (28) can be written as

$$\min_{z \in \Lambda} f(z), \quad (41)$$

where

$$\Lambda = \{z \in \mathbb{R}^{3n_1 n_2 n_3 + 4\|\mathbf{r}\|_1(n_1 + n_2)} \mid P_\Omega(\mathcal{C}_i \mathbf{i} + \mathcal{C}_j \mathbf{j} + \mathcal{C}_k \mathbf{k} - \mathcal{M}) = 0\}.$$

Therefore, the projected gradient of f at $z \in \Lambda$ is given as

$$\Pi_\Lambda(\nabla f(z)) = \begin{pmatrix} \left[P_{\Omega^c}\left(\frac{\partial f(z)}{\partial \mathcal{C}_i}\right); P_{\Omega^c}\left(\frac{\partial f(z)}{\partial \mathcal{C}_j}\right); P_{\Omega^c}\left(\frac{\partial f(z)}{\partial \mathcal{C}_k}\right) \right] \\ \left[\frac{\partial f(z)}{\partial \hat{\mathcal{A}}_e}; \frac{\partial f(z)}{\partial \hat{\mathcal{A}}_i}; \frac{\partial f(z)}{\partial \hat{\mathcal{A}}_j}; \frac{\partial f(z)}{\partial \hat{\mathcal{A}}_k} \right] \\ \left[\frac{\partial f(z)}{\partial \hat{\mathcal{B}}_e}; \frac{\partial f(z)}{\partial \hat{\mathcal{B}}_i}; \frac{\partial f(z)}{\partial \hat{\mathcal{B}}_j}; \frac{\partial f(z)}{\partial \hat{\mathcal{B}}_k} \right] \end{pmatrix},$$

where $\Pi_\Lambda(\cdot)$ denotes the projection onto the feasible set Λ .

Definition 4.2 (stationary point). The point $z^* \in \Lambda$ is said to be a stationary point of the low gQt-rank quaternion tensor completion model (28) if $\Pi_\Lambda(\nabla f(z^*)) = 0$.

The following theorems show that the sequence generated by Algorithm 2 is bounded and any accumulation point converges to a stationary point of (28).

Theorem 4.2. Let $\{z^t\}$ be the sequence generated by QRTC. Then, there exists a constant K_1 such that

$$f(z^t) - f(z^{t+1}) \geq K_1 \|z^t - z^{t+1}\|_2^2. \quad (42)$$

Moreover, the sequence $\{z^t\}$ is bounded.

Proof. Let h_1 be defined as (33). It is easy to see that $f(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k \mid \hat{\mathcal{A}}, \hat{\mathcal{B}}) + h_1(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k) - \frac{1}{2} \|P_{\Omega^c}(\Re(\mathcal{C}))\|_F^2$ is a convex function. Since h_1 is an indicator function on affine space, we have

$$\frac{\partial(f + h_1)}{\partial \mathcal{C}_\alpha}(\cdot) = P_{\Omega^c}\left(\frac{\partial f}{\partial \mathcal{C}_\alpha}(\cdot)\right), \quad \alpha = \mathbf{i}, \mathbf{j}, \mathbf{k}.$$

It follows from

$$h_1(\mathcal{C}_i^{t+1}, \mathcal{C}_j^{t+1}, \mathcal{C}_k^{t+1}) = 0, \quad h_1(\mathcal{C}_i^t, \mathcal{C}_j^t, \mathcal{C}_k^t) = 0, \quad \|\mathcal{C}^{t+1} - \mathcal{C}^t\|_F = \|P_{\Omega^c}(\mathcal{C}^{t+1} - \mathcal{C}^t)\|_F, \quad (43)$$

and the convexity of function $f(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k | \hat{\mathcal{A}}, \hat{\mathcal{B}}) + h_1(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k) - \frac{1}{2}\|P_{\Omega^c}(\mathfrak{R}(\mathcal{C}))\|_F^2$ that

$$\begin{aligned} f(\mathcal{C}^t | \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t) - f(\mathcal{C}^{t+1} | \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t) &= f(\mathcal{C}_i^t, \mathcal{C}_j^t, \mathcal{C}_k^t | \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t) - f(\mathcal{C}_i^{t+1}, \mathcal{C}_j^{t+1}, \mathcal{C}_k^{t+1} | \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t) \\ &\geq \sum_{\alpha=i,j,k} \left(\left\langle \frac{\partial f + h_1}{\partial \mathcal{C}_\alpha}(\mathcal{C}_i^{t+1}, \mathcal{C}_j^{t+1}, \mathcal{C}_k^{t+1} | \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t), \mathcal{C}_\alpha^t - \mathcal{C}_\alpha^{t+1} \right\rangle + \frac{1}{2} \|\mathcal{C}_\alpha^t - \mathcal{C}_\alpha^{t+1}\|_F^2 \right) \\ &= \sum_{\alpha=i,j,k} \left\langle P_{\Omega^c} \left(\frac{\partial f}{\partial \mathcal{C}_\alpha}(\mathcal{C}_i^{t+1}, \mathcal{C}_j^{t+1}, \mathcal{C}_k^{t+1} | \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t) \right), P_{\Omega^c}(\mathcal{C}_\alpha^t - \mathcal{C}_\alpha^{t+1}) \right\rangle \\ &\quad + \frac{1}{2} \|\mathcal{C}^t - \mathcal{C}^{t+1}\|_F^2 \end{aligned} \quad (44)$$

From (39),

$$P_{\Omega^c}(\delta_\alpha^{t+1}) = P_{\Omega^c} \left(\frac{\partial f}{\partial \mathcal{C}_\alpha}(\mathcal{C}_i^{t+1}, \mathcal{C}_j^{t+1}, \mathcal{C}_k^{t+1} | \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t) \right), \quad \alpha = i, j, k. \quad (45)$$

With (44), we can obtain

$$\begin{aligned} f(\mathcal{C}^t | \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t) - f(\mathcal{C}^{t+1} | \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t) &\geq \sum_{\alpha=i,j,k} \langle P_{\Omega^c}(\delta_\alpha^{t+1}), P_{\Omega^c}(\mathcal{C}_\alpha^t - \mathcal{C}_\alpha^{t+1}) \rangle + \frac{1}{2} \|\mathcal{C}^t - \mathcal{C}^{t+1}\|_F^2 \\ &\geq -\|P_{\Omega^c}(\delta^{t+1})\|_F \|P_{\Omega^c}(\mathcal{C}^t - \mathcal{C}^{t+1})\|_F + \frac{1}{2} \|\mathcal{C}^t - \mathcal{C}^{t+1}\|_F^2 \\ &\geq \frac{1}{4} \|\mathcal{C}^t - \mathcal{C}^{t+1}\|_F^2, \end{aligned} \quad (46)$$

where the last inequality holds due to (43) and (40). Thus,

$$\begin{aligned} f(\hat{\mathcal{A}}^t | \mathcal{C}^{t+1}, \hat{\mathcal{B}}^t) - f(\hat{\mathcal{A}}^{t+1} | \mathcal{C}^{t+1}, \hat{\mathcal{B}}^t) &= \frac{1}{n_3} \sum_{l=1}^{n_3} \left(\frac{1}{2} \|\hat{\mathcal{A}}^{t,(l)} \hat{\mathcal{B}}^{t,(l)} - \hat{\mathcal{C}}^{t+1,(l)}\|_F^2 + \frac{\lambda}{2} \|\hat{\mathcal{A}}^{t,(l)}\|_F^2 \right. \\ &\quad \left. - \frac{1}{2} \|\hat{\mathcal{A}}^{t+1,(l)} \hat{\mathcal{B}}^{t,(l)} - \hat{\mathcal{C}}^{t+1,(l)}\|_F^2 - \frac{\lambda}{2} \|\hat{\mathcal{A}}^{t+1,(l)}\|_F^2 \right) \\ &= \frac{1}{n_3} \sum_{l=1}^{n_3} \left(\frac{1}{2} \|(\hat{\mathcal{A}}^{t,(l)} - \hat{\mathcal{A}}^{t+1,(l)}) \hat{\mathcal{B}}^{t,(l)}\|_F^2 + \frac{\lambda}{2} \|\hat{\mathcal{A}}^{t,(l)} - \hat{\mathcal{A}}^{t+1,(l)}\|_F^2 \right. \\ &\quad \left. \Re \left(\text{tr}((\hat{\mathcal{A}}^{t+1,(l)} \hat{\mathcal{B}}^{t,(l)} - \hat{\mathcal{C}}^{t+1,(l)})(\hat{\mathcal{B}}^{t,(l)})^* (\hat{\mathcal{A}}^{t,(l)} - \hat{\mathcal{A}}^{t+1,(l)})^*) \right) \right. \\ &\quad \left. + \lambda \text{tr}(\hat{\mathcal{A}}^{t+1,(l)} (\hat{\mathcal{A}}^{t,(l)} - \hat{\mathcal{A}}^{t+1,(l)})^*) \right) \Big). \end{aligned} \quad (47)$$

By Theorem 4.1 and (37), it holds for any $l \in [n_3]$,

$$(\hat{\mathcal{A}}^{t+1,(l)} \hat{\mathcal{B}}^{t,(l)} - \hat{\mathcal{C}}^{t+1,(l)})(\hat{\mathcal{B}}^{t,(l)})^* + \lambda \hat{\mathcal{A}}^{t+1,(l)} = \beta(\hat{\mathcal{A}}^{t,(l)} - \hat{\mathcal{A}}^{t+1,(l)}),$$

which, together with (47), implies

$$\begin{aligned} f(\hat{\mathcal{A}}^t | \mathcal{C}^{t+1}, \hat{\mathcal{B}}^t) - f(\hat{\mathcal{A}}^{t+1} | \mathcal{C}^{t+1}, \hat{\mathcal{B}}^t) &= \frac{1}{n_3} \sum_{l=1}^{n_3} \left(\frac{1}{2} \|(\hat{\mathcal{A}}^{t,(l)} - \hat{\mathcal{A}}^{t+1,(l)}) \hat{\mathcal{B}}^{t,(l)}\|_F^2 + \left(\frac{\lambda}{2} + \beta \right) \|\hat{\mathcal{A}}^{t,(l)} - \hat{\mathcal{A}}^{t+1,(l)}\|_F^2 \right) \\ &\geq \frac{1}{n_3} \sum_{l=1}^{n_3} \left(\frac{\lambda}{2} + \beta \right) \|\hat{\mathcal{A}}^{t,(l)} - \hat{\mathcal{A}}^{t+1,(l)}\|_F^2 \\ &= \frac{\lambda + 2\beta}{2n_3} \|\hat{\mathcal{A}}^t - \hat{\mathcal{A}}^{t+1}\|_F^2. \end{aligned} \quad (48)$$

Similarly, we have

$$f(\hat{\mathcal{B}}^t | \mathcal{C}^{t+1}, \hat{\mathcal{A}}^{t+1}) - f(\hat{\mathcal{B}}^{t+1} | \mathcal{C}^{t+1}, \hat{\mathcal{A}}^{t+1}) \geq \frac{\lambda + 2\beta}{2n_3} \|\hat{\mathcal{B}}^t - \hat{\mathcal{B}}^{t+1}\|_F^2. \quad (49)$$

Therefore, combining (46), (48) and (49), we get

$$f(\mathcal{C}^t, \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t) - f(\mathcal{C}^{t+1}, \hat{\mathcal{A}}^{t+1}, \hat{\mathcal{B}}^{t+1}) \geq \min\left(\frac{1}{4}, \frac{\lambda + 2\beta}{2n_3}\right) \|z^t - z^{t+1}\|_2^2.$$

Taking

$$K_1 = \min\left(\frac{1}{4}, \frac{\lambda}{2} + \frac{\lambda + 2\beta}{2n_3}\right),$$

we prove that (42) holds and hence the sequence $\{f(z^t)\}$ is monotonically decreasing.

It follows from $f \geq 0$ that

$$\sum_{t=1}^{\infty} (f(z^t) - f(z^{t+1})) < \infty, \quad \sum_{t=1}^{\infty} \|z^t - z^{t+1}\|_2^2 < \infty, \quad \lim_{t \rightarrow \infty} z^t - z^{t+1} = 0.$$

Since

$$f(z^1) \geq f(z^t) \geq \frac{\lambda}{2} (\|\mathcal{A}^t\|_F^2 + \|\mathcal{B}^t\|_F^2),$$

$\{\mathcal{A}^t\}$, $\{\mathcal{B}^t\}$ are bounded, and so $\{\hat{\mathcal{A}}^t\}$, $\{\hat{\mathcal{B}}^t\}$ are also bounded. Together with the fact that

$$f(z^1) \geq f(z^t) \geq \|\mathcal{A}^t *_{\mu} \mathcal{B}^t - \mathcal{C}^t\|_F^2,$$

$\{\mathcal{C}^t\}$ is also bounded, and hence $\{z^t\}$ is bounded. \square

Theorem 4.3. *Let $\{z^t\}$ be the sequence generated by QRTC. Then, there exists a constant K_2 such that*

$$K_2 \|z^t - z^{t+1}\|_2 \geq \|\Pi_{\Lambda}(\nabla f(z^t))\|_F. \quad (50)$$

Moreover, any accumulation point of $\{z^t\}$ is a stationary point of (28).

Proof. By Theorem 4.2, $\{z^t\}$ is bounded. So, there exists a compact convex set Z such that $\{z^t\} \subset Z$. Since f is a quadratic polynomial in Z , its gradient is Lipschitz in Z with the Lipschitz constant L_f , that is,

$$\|\nabla f(z) - \nabla f(z')\|_2 \leq L_f \|z - z'\|_2, \quad \forall z, z' \in Z.$$

By Theorem 4.1, (37) and (38), for any $l \in [n_3]$, we have

$$\begin{aligned} \nabla_{\mathbb{H}} f(\hat{\mathcal{A}}^{t+1, (l)} | \hat{\mathcal{A}}^{t+1, (-l)}, \mathcal{C}^{t+1}, \hat{\mathcal{B}}^t) &= \frac{\beta}{n_3} (\hat{\mathcal{A}}^{t, (l)} - \hat{\mathcal{A}}^{t+1, (l)}), \\ \nabla_{\mathbb{H}} f(\hat{\mathcal{B}}^{t+1, (l)} | \hat{\mathcal{B}}^{t+1, (-l)}, \mathcal{C}^{t+1}, \hat{\mathcal{A}}^{t+1}) &= \frac{\beta}{n_3} (\hat{\mathcal{B}}^{t, (l)} - \hat{\mathcal{B}}^{t+1, (l)}), \end{aligned}$$

where $\hat{\mathcal{A}}^{t+1, (-l)}$ and $\hat{\mathcal{B}}^{t+1, (-l)}$ denote $\hat{\mathcal{A}}^{t+1}$ and $\hat{\mathcal{B}}^{t+1}$ except $\hat{\mathcal{A}}^{t+1, (l)}$ and $\hat{\mathcal{B}}^{t+1, (l)}$, respectively. Set

$$\frac{\partial f}{\partial \hat{\mathcal{A}}} = \left[\frac{\partial f}{\partial \hat{\mathcal{A}}_{\mathbf{e}}}, \frac{\partial f}{\partial \hat{\mathcal{A}}_{\mathbf{i}}}, \frac{\partial f}{\partial \hat{\mathcal{A}}_{\mathbf{j}}}, \frac{\partial f}{\partial \hat{\mathcal{A}}_{\mathbf{k}}} \right], \quad \frac{\partial f}{\partial \hat{\mathcal{B}}} = \left[\frac{\partial f}{\partial \hat{\mathcal{B}}_{\mathbf{e}}}, \frac{\partial f}{\partial \hat{\mathcal{B}}_{\mathbf{i}}}, \frac{\partial f}{\partial \hat{\mathcal{B}}_{\mathbf{j}}}, \frac{\partial f}{\partial \hat{\mathcal{B}}_{\mathbf{k}}} \right],$$

then,

$$\begin{aligned} \left\| \frac{\partial f}{\partial \hat{\mathcal{A}}}(z^t) \right\|_F &\leq \left\| \frac{\partial f}{\partial \hat{\mathcal{A}}}(z^t) - \frac{\partial f}{\partial \hat{\mathcal{A}}}(\mathcal{C}^{t+1}, \mathcal{A}^{t+1}, \mathcal{B}^t) \right\|_F + \left\| \frac{\partial f}{\partial \hat{\mathcal{A}}}(\mathcal{C}^{t+1}, \mathcal{A}^{t+1}, \mathcal{B}^t) \right\|_F \\ &\leq L_f \|z^t - z^{t+1}\|_2 + \sum_{l=1}^{n_3} \left\| \frac{\partial f}{\partial \hat{\mathcal{A}}^{(l)}}(\mathcal{C}^{t+1}, \mathcal{A}^{t+1}, \mathcal{B}^t) \right\|_F \\ &= L_f \|z^t - z^{t+1}\|_2 + \sum_{l=1}^{n_3} \left\| \nabla_{\mathbb{H}} f(\hat{\mathcal{A}}^{t+1, (l)} | \hat{\mathcal{A}}^{t+1, (-l)}, \mathcal{C}^{t+1}, \hat{\mathcal{B}}^t) \right\|_F \\ &= L_f \|z^t - z^{t+1}\|_2 + \sum_{l=1}^{n_3} \left\| \frac{\beta}{n_3} (\hat{\mathcal{A}}^{t, (l)} - \hat{\mathcal{A}}^{t+1, (l)}) \right\|_F \\ &\leq (L_f + \frac{\beta}{n_3}) \|z^t - z^{t+1}\|_2. \end{aligned} \quad (51)$$

Similarly, we have

$$\left\| \frac{\partial f}{\partial \hat{\mathcal{B}}}(z^t) \right\|_F \leq (L_f + \frac{\beta}{n_3}) \|z^t - z^{t+1}\|_2. \quad (52)$$

It follows from (45) that

$$\begin{aligned} \sum_{\alpha=\mathbf{i}, \mathbf{j}, \mathbf{k}} \|P_{\Omega^c} \frac{\partial f}{\partial \mathcal{C}_\alpha}(z^t)\|_F &\leq \sum_{\alpha=\mathbf{i}, \mathbf{j}, \mathbf{k}} \left(\left\| (P_{\Omega^c} \frac{\partial f}{\partial \mathcal{C}_\alpha}(z^t) - P_{\Omega^c} \frac{\partial f}{\partial \mathcal{C}_\alpha}(\mathcal{C}^{t+1}, \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t)) \right\|_F + \|P_{\Omega^c} \frac{\partial f}{\partial \mathcal{C}_\alpha}(\mathcal{C}^{t+1}, \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t)\|_F \right) \\ &\leq \|\Pi_\Lambda(\nabla f(z^t)) - \Pi_\Lambda(\nabla f(\mathcal{C}^{t+1}, \hat{\mathcal{A}}^t, \hat{\mathcal{B}}^t))\|_F + \sum_{\alpha=\mathbf{i}, \mathbf{j}, \mathbf{k}} \|P_{\Omega^c}(\delta_\alpha^{t+1})\|_F \\ &\leq L_f \|z^t - z^{t+1}\|_2 + \frac{1}{4} \|\mathcal{C}^t - \mathcal{C}^{t+1}\|_F \\ &\leq (L_f + \frac{1}{4}) \|z^t - z^{t+1}\|_2. \end{aligned} \quad (53)$$

Combining (51), (52) and (53), it is easy to see that (50) holds with $K_2 = L_f + \max(\frac{1}{4}, \frac{\beta}{n_3})$.

Since $\{z^t\}$ is bounded, there exists a convergent subsequence of $\{z^t\}$. Without loss of generality, we assume that $\lim_{k \rightarrow \infty} z^{t_k} = z^*$. Then,

$$\begin{aligned} \|\Pi_\Lambda(\nabla f(z^*))\|_F &\leq \|\Pi_\Lambda(\nabla f(z^*) - \nabla f(z^{t_k}))\|_F + \|\Pi_\Lambda(\nabla f(z^{t_k}))\|_F \\ &\leq L_f \|z^* - z^{t_k}\|_2 + K_3 \|z^{t_k} - z^{t_k+1}\|_2, \end{aligned}$$

which, together with taking limit $k \rightarrow \infty$ in the right hand side, shows $\Pi_\Lambda(\nabla f(z^*)) = 0$, and hence z^* is a stationary point of (28). \square

Theorems 4.2 and 4.3 show that the sequence $\{z^t\}$ generated by QRTC is bounded and its any accumulation point is a stationary point of (28). We next use the Kurdyka-Łojasiewicz (KL) property [38, 39, 40] to prove that $\{z^t\}$ is convergent.

Definition 4.3 (KL property). Let $Z \in \mathbb{R}^n$ be an open set and $f : Z \rightarrow \mathbb{R}$ be a semi-algebraic function. For every critical point $z^* \in Z$ of f , there is a neighborhood $Z' \in Z$ of z^* , an exponent $\theta \in [0, 1)$ and a positive constant K_3 such that

$$|f(z) - f(z^*)|^\theta \leq K_3 \|\Pi_\Lambda(\nabla f(z))\|_F, \quad \forall z \in Z'. \quad (54)$$

It is obvious that $f(z) + \delta_\Lambda(z)$ is a semi-algebraic function. Then, from Definition 4.3 and [40], the KL inequality (54) holds for the function $f(z) + \delta_\Lambda(z)$. Hence, for the given θ, K_3 in Definition 4.3 and K_1, K_2 are defined as Theorems 4.2 and 4.3, we have the following convergence results.

Theorem 4.4. Let $\{z^t\}$ be the sequence generated by QRTC and z^* be an accumulation point of $\{z^t\}$. Assume $z^0 \in B(z^*, \tau) \doteq \{z \mid \|z - z^*\|_F < \tau\} \subset Z'$ with

$$\tau > \frac{K_2 K_3}{K_1(1-\theta)} |f(z^0) - f(z^*)|^{1-\theta} + \|z^0 - z^*\|_2,$$

then, $z^t \in B(z^*, \tau)$ for $t = 0, 1, 2, \dots$. Moreover,

$$\sum_{t=0}^{\infty} \|z^{t+1} - z^t\|_2 \leq \frac{K_2 K_3}{K_1(1-\theta)} |f(z^0) - f(z^*)|^{1-\theta}, \quad \lim_{t \rightarrow \infty} z^t = z^*. \quad (55)$$

Proof. We show $\{z^t\} \subset B(z^*, \tau)$ by induction. When $t = 0$, it holds obviously. Assume that $z^t \in B(z^*, \tau)$ holds for all $t \leq \hat{t}$, then KL property is true for z^t . Now we display that $z^t \in B(z^*, \tau)$ is true when $t = \hat{t} + 1$. Let

$$\phi(x) \doteq \frac{K_3}{1-\theta} |x - f(z^*)|^{1-\theta}, \quad x > f(z^*).$$

Then $\phi(x)$ is concave and differentiable. Hence, we have

$$\phi(f(z^t)) - \phi(f(z^{t+1})) \geq \phi'(f(z^t))(f(z^t) - f(z^{t+1})) = \frac{K_3}{|f(z^t) - f(z^*)|^\theta} (f(z^t) - f(z^{t+1})).$$

By Definition 4.3, (42) and (50), we get

$$\phi(f(z^t)) - \phi(f(z^{t+1})) \geq \frac{1}{\|\Pi_\Lambda(\nabla f(z^t))\|_F} (f(z^t) - f(z^{t+1})) \geq \frac{K_1}{K_2} \|z^t - z^{t+1}\|_2.$$

Therefore,

$$\sum_{p=0}^t \|z^p - z^{p+1}\|_2 \leq \frac{K_2}{K_1} \sum_{p=0}^t (\phi(f(z^p)) - \phi(f(z^{p+1}))) \leq \frac{K_2}{K_1} \phi(f(z^0)), \quad (56)$$

which implies

$$\|z^{\hat{t}+1} - z^*\|_2 \leq \sum_{p=0}^{\hat{t}} \|z^{p+1} - z^p\|_2 + \|z^0 - z^*\|_2 \leq \frac{K_2}{K_1} \phi(f(z^0)) + \|z^0 - z^*\|_2 \leq \tau.$$

Thus, $\{z^t\} \subset B(z^*, \tau)$.

Taking $t \rightarrow \infty$ in (56), the first inequality in (55) is arrived. Without loss of generality, we assume that $\lim_{k \rightarrow \infty} z^{t_k} = z^*$. Then, for all $t > 0$ and $t_{k+1} \geq t > t_k$,

$$\|z^t - z^*\|_2 \leq \|z^{t_k} - z^*\|_2 + \sum_{p=t_k}^{t-1} \|z^{p+1} - z^p\|_2,$$

which, together with the fact that $\{\|z^{t+1} - z^t\|_2\}$ is Cauchy sequence, implies $\lim_{t \rightarrow \infty} z^t = z^*$. \square

Theorem 4.5. Suppose that $\{z^t\}$ is the sequence generated by QRTC and z^* be its limit point. Then, the following statements hold.

(i) If $\theta \in (0, \frac{1}{2}]$, there exist $\gamma > 0$ and $\xi \in (0, 1)$ such that

$$\|z^t - z^*\|_2 \leq \gamma \xi^t.$$

(ii) If $\theta \in (\frac{1}{2}, 1)$, there exists $\gamma > 0$ such that

$$\|z^t - z^*\|_2 \leq \gamma t^{-\frac{1-\theta}{2\theta-1}}.$$

Proof. Since $\{z^t\}$ converges to z^* , there exists an index k_0 such that $z^{k_0} \in B(z^*, \tau)$, where τ is given in Theorem 4.4. Hence, we can regard z^{k_0} as an initial point. Without loss of generality, we set $z^0 \in B(z^*, \tau)$. Let

$$\Delta_t \doteq \sum_{p=t}^{\infty} \|z^p - z^{p+1}\|_2 \geq \|z^t - z^*\|_2. \quad (57)$$

It follows from (56) that

$$\Delta_t \leq \frac{K_2}{K_1} \phi(f(z^t)) = \frac{K_2 K_3}{K_1(1-\theta)} |f(z^t) - f(z^*)|^{1-\theta} = \frac{K_2 K_3}{K_1(1-\theta)} (|f(z^t) - f(z^*)|^\theta)^{\frac{1-\theta}{\theta}}.$$

Using KL inequality (54), it holds

$$\Delta_t \leq \frac{K_2 K_3}{K_1(1-\theta)} (K_3 \|\Pi_\Lambda(\nabla f(z^t))\|_F)^{\frac{1-\theta}{\theta}}.$$

Set $\xi_1 = \frac{(K_2 K_3)^{\frac{1}{\theta}}}{K_1(1-\theta)}$, then the above inequality, together with (42) and (50), implies

$$\Delta_t \leq \frac{K_2 K_3}{K_1(1-\theta)} (K_2 K_3 \|z^t - z^{t+1}\|_2)^{\frac{1-\theta}{\theta}} = \xi_1 (\Delta_t - \Delta_{t+1})^{\frac{1-\theta}{\theta}}. \quad (58)$$

We now prove (i). If $\theta \in (0, \frac{1}{2}]$, then $\frac{1-\theta}{\theta} \geq 1$. It holds for sufficiently large t ,

$$\Delta_t \leq \xi_1 (\Delta_t - \Delta_{t+1}).$$

Hence,

$$\Delta_{t+1} \leq \frac{\xi_1 - 1}{\xi_1},$$

which together with (57) implies that (i) holds with $\xi = \frac{\xi_1 - 1}{\xi_1}$.

We next to prove (ii). If $\theta \in (\frac{1}{2}, 1)$, let $q(x) = x^{-\frac{\theta}{1-\theta}}$. Then, $q(x)$ is monotonically decreasing on x . It follows from (58) that

$$\xi_1^{-\frac{\theta}{1-\theta}} \leq q(\Delta_t)(\Delta_t - \Delta_{t+1}) = \int_{\Delta_t}^{\Delta_{t+1}} q(\Delta_t) dx \leq \int_{\Delta_t}^{\Delta_{t+1}} q(x) dx = -\frac{1-\theta}{2\theta-1} \left(\Delta_t^{-\frac{2\theta-1}{1-\theta}} - \Delta_{t+1}^{-\frac{2\theta-1}{1-\theta}} \right).$$

Define $\nu \doteq -\frac{2\theta-1}{1-\theta}$. Then $\nu < 0$ and hence

$$\Delta_{t+1}^\nu - \Delta_t^\nu \geq -\nu \xi_1^{-\frac{\theta}{1-\theta}} > 0.$$

Thus, there exists \bar{t} such that for all $t \geq 2\bar{t}$,

$$\Delta_t^\nu \geq \Delta_{\bar{t}}^\nu - \nu \xi_1^{-\frac{\theta}{1-\theta}} (t - \bar{t}) \geq -\frac{\nu}{2} \xi_1^{-\frac{\theta}{1-\theta}} t,$$

which implies

$$\Delta_t \leq \gamma t^{\frac{1}{\nu}}.$$

Let $\gamma = \left(-\frac{\nu}{2} \xi_1^{-\frac{\theta}{1-\theta}} \right)^{\frac{1}{\nu}}$. Then, the above inequality shows that (ii) holds. \square

4.4 Low multi-gQt-rank tensor completion model

In this subsection, we establish a novel low-rank tensor completion model based on multi-gQt-rank and then present the tensor factorization based solution method.

Different from (28), we replace the loss function in mode-3, i.e., $\frac{1}{2} \|\mathcal{A} *_{\mu} \mathcal{B} - \mathcal{C}\|_F^2$, with a weighted sum of the loss functions in three modes, and consider the following model

$$\min_{\mathcal{C}, \mathcal{A}_w, \mathcal{B}_w, w \in [3]} g(\mathcal{C}, \mathcal{A}_1, \mathcal{B}_1, \mathcal{A}_2, \mathcal{B}_2, \mathcal{A}_3, \mathcal{B}_3), \quad \text{s.t. } P_{\Omega}(\mathcal{C} - \mathcal{M}) = 0, \quad \Re(\mathcal{C}) = 0, \quad (59)$$

where,

$$g(\mathcal{C}, \mathcal{A}_1, \mathcal{B}_1, \mathcal{A}_2, \mathcal{B}_2, \mathcal{A}_3, \mathcal{B}_3) = \sum_{w=1}^3 \left(\frac{\alpha_w}{2} \|\mathcal{A}_w *_{\mu}^w \mathcal{B}_w - \mathcal{C}\|_F^2 + \frac{\lambda}{2} (\|\mathcal{A}_w\|_F^2 + \|\mathcal{B}_w\|_F^2) \right) + \sum_{k=1}^2 \lambda_k \|\mathcal{C} \times_k H_{n_k}\|_F^2,$$

and α_w ($w = 1, 2, 3$) is the weighted coefficient.

Similar to Subsection 4.2, we solve (59) as the following steps. First, update \mathcal{C}^t by

$$\begin{aligned} & \arg \min_{\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k} \sum_{w=1}^3 \frac{\alpha_w}{2} \|\mathcal{A}_w^{t-1} *_{\mu}^w \mathcal{B}_w^{t-1} - \mathcal{C}\|_F^2 + \sum_{k=1}^2 \lambda_k \|\mathcal{C} \times_k H_{n_k}\|_F^2 + \delta_{\{P_{\Omega}(\mathcal{C} - \mathcal{M})=0\}}(\mathcal{C}) \\ & = \arg \min_{\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k} \frac{1}{2} \sum_{w=1}^3 \alpha_w \|\mathcal{X}^{t-1} - \mathcal{C}\|_F^2 + \sum_{k=1}^2 \lambda_k \|\mathcal{C} \times_k H_{n_k}\|_F^2 + \delta_{\{P_{\Omega}(\mathcal{C} - \mathcal{M})=0\}}(\mathcal{C}), \end{aligned} \quad (60)$$

where $\mathcal{X}^{t-1} = \frac{1}{\alpha_1 + \alpha_2 + \alpha_3} \sum_{w=1}^3 \alpha_w \mathcal{A}_w^{t-1} *_{\mu}^w \mathcal{B}_w^{t-1}$. Let

$$\begin{aligned} f_2(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k) &= \sum_{w=1}^3 \frac{\alpha_w}{2} \|\Im(\mathcal{X}^{t-1}) - \mathcal{C}\|_F^2 + \sum_{k=1}^2 \lambda_k \|\mathcal{C} \times_k H_{n_k}\|_F^2, \\ h_2(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k) &= \delta_{\{P_{\Omega}(\mathcal{C} - \mathcal{M})=0\}}(\mathcal{C}). \end{aligned}$$

Similar to Algorithm 1, we can solve (60) by the following PGM.

Algorithm 3 PGM for (60)

Input. The tensor data $\mathcal{M}, \mathcal{X}^{t-1}$, the observed set Ω , initial step size α_0 , parameters $\lambda, \lambda_i, H_{n_i}, 1 \leq i \leq 3$.

Step 0. Initialize $z^0 = [\mathcal{C}_i^0, \mathcal{C}_j^0, \mathcal{C}_k^0]$ which satisfies $P_\Omega(\mathcal{C}_i^0 \mathbf{i} + \mathcal{C}_j^0 \mathbf{j} + \mathcal{C}_k^0 \mathbf{k} - \mathcal{M}) = 0$, $z^{-1} = z^0$. Iterate the following steps for $k = 0, 1, 2, \dots$ while it does not satisfy stop criterion.

Step 1. If $k \geq 1$, choose step size

$$\alpha_k = \frac{\|z^k - z^{k-1}\|_F^2}{\langle z^k - z^{k-1}, \nabla f_2(z^k) - \nabla f_2(z^{k-1}) \rangle}, \text{ or } \alpha_k = \frac{\langle z^k - z^{k-1}, \nabla f_2(z^k) - \nabla f_2(z^{k-1}) \rangle}{\|\nabla f_2(z^k) - \nabla f_2(z^{k-1})\|_F^2}.$$

Step 2. Update $[\mathcal{C}_i^{k+1}, \mathcal{C}_j^{k+1}, \mathcal{C}_k^{k+1}] = \text{prox}_{\alpha_k h_2}([\mathcal{C}_i^k, \mathcal{C}_j^k, \mathcal{C}_k^k] - \alpha_k \nabla f_2(\mathcal{C}_i^k, \mathcal{C}_j^k, \mathcal{C}_k^k))$.

Output. $\mathcal{C}_i^{k+1} \mathbf{i} + \mathcal{C}_j^{k+1} \mathbf{j} + \mathcal{C}_k^{k+1} \mathbf{k}$.

It follows from Theorem 3.7 that (59) is rewritten as

$$\min_{\hat{\mathcal{C}}, \hat{\mathcal{A}}_w, \hat{\mathcal{B}}_w, w \in [3]} \sum_{w=1}^3 \sum_{l=1}^{n_w} \left(\frac{\alpha_w}{2n_w} \|\hat{\mathcal{A}}_w^{(l)} \hat{\mathcal{B}}_w^{(l)} - \hat{\mathcal{C}}_w^{(l)}\|_F^2 + \frac{\lambda}{2n_w} (\|\mathcal{A}_w^{(l)}\|_F^2 + \|\mathcal{B}_w^{(l)}\|_F^2) \right) + \sum_{k=1}^2 \lambda_k \|\mathcal{C} \times_k H_{n_k}\|_F^2.$$

To update $\hat{\mathcal{A}}_w^{t,(l)}$ and $\hat{\mathcal{B}}_w^{t,(l)}$, we consider the following problem

$$\min_{\hat{\mathcal{A}}_w^{t,(l)}, \hat{\mathcal{B}}_w^{t,(l)}} \frac{\alpha_w}{2n_w} \|\hat{\mathcal{A}}_w^{(l)} \hat{\mathcal{B}}_w^{(l)} - \hat{\mathcal{C}}_w^{(l)}\|_F^2 + \frac{\lambda}{2n_w} (\|\mathcal{A}_w^{(l)}\|_F^2 + \|\mathcal{B}_w^{(l)}\|_F^2).$$

For $l \in [n_3]$ and $w \in [3]$, update $\hat{\mathcal{A}}_w^{(l)}$ by

$$\begin{aligned} \hat{\mathcal{A}}_w^{t,(l)} &= \arg \min_{\hat{\mathcal{A}}_w} \frac{\alpha_w}{2} \|\hat{\mathcal{A}}_w^{(l)} \hat{\mathcal{B}}_w^{t-1,(l)} - \hat{\mathcal{C}}_w^{t,(l)}\|_F^2 + \frac{\lambda}{2} \|\hat{\mathcal{A}}_w^{(l)}\|_F^2 + \frac{\beta}{2} \|\hat{\mathcal{A}}_w^{(l)} - \hat{\mathcal{A}}_w^{t-1,(l)}\|_F^2 \\ &= (\alpha_w \hat{\mathcal{C}}_w^{t,(l)} (\hat{\mathcal{B}}_w^{t-1,(l)})^* + \beta \hat{\mathcal{A}}_w^{t-1,(l)}) (\alpha_w \hat{\mathcal{B}}_w^{t-1,(l)} (\hat{\mathcal{B}}_w^{t-1,(l)})^* + (\lambda + \beta) I)^{-1}, \end{aligned} \quad (61)$$

and the updating of $\hat{\mathcal{B}}_w^{(l)}$ is given by

$$\begin{aligned} \hat{\mathcal{B}}_w^{t,(l)} &= \arg \min_{\hat{\mathcal{B}}_w} \frac{\alpha_w}{2} \|\hat{\mathcal{A}}_w^{t,(l)} \hat{\mathcal{B}}_w^{(l)} - \hat{\mathcal{C}}_w^{t,(l)}\|_F^2 + \frac{\lambda}{2} \|\hat{\mathcal{B}}_w^{(l)}\|_F^2 + \frac{\beta}{2} \|\hat{\mathcal{B}}_w^{(l)} - \hat{\mathcal{B}}_w^{t-1,(l)}\|_F^2 \\ &= (\alpha_w (\hat{\mathcal{A}}_w^{t,(l)})^* \hat{\mathcal{A}}_w^{t,(l)} + (\lambda + \beta) I)^{-1} (\alpha_w (\hat{\mathcal{A}}_w^{t,(l)})^* \hat{\mathcal{C}}_w^{t,(l)} + \beta \hat{\mathcal{B}}_w^{t-1,(l)}). \end{aligned} \quad (62)$$

Consequently, we propose the following algorithm to solve (59). The convergence analysis of Algorithm 4 is similar to that of Algorithm 2 and hence we omit it here.

5 Numerical Experiments

In this section, we report some numerical results of our proposed algorithms QRTC and MQRTC to show the validity. Moreover, we compare them with several existing state-of-the-art methods, including TMac [19], TNN [4], TCTF [3] and LRQA-2 [34]. Note that TMac has two versions, i.e., TMac-dec and TMac-inc, and the former uses the rank-decreasing scheme to adjust its rank while the latter employs the rank-increasing scheme. The codes of TMac³, TNN⁴, TCTF⁵ are open source and LRQA-2 is provided by the authors in [34].

A color video is a 4-way tensor defined by two indices for spatial variables, one index for temporal variable and one index for color mode. All the videos in our simulations are initially represented by 4-way tensors $\mathcal{C} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times 3}$, where $n_1 \times n_2$ stands for the pixel scale of each frame, and n_3 is the number of

³<https://xu-yangyang.github.io/codes/TMac.zip>

⁴http://www.ece.tufts.edu/~shuchin/tensor_completion_and_rpca.zip

⁵https://panzhous.github.io/assets/code/TCTF_code.rar

Algorithm 4 Multi gQt-Rank Tensor Completion (MQRTC)

Input. The tensor data $\mathcal{M} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$, the observed set Ω , the rank $\mathbf{r} \in \mathbb{Z}_+^{n_3}$, parameters $\lambda, \lambda_i, H_{n_i}, 1 \leq i \leq 2$ and ϵ .

Step 0. Initialize $\hat{\mathcal{A}}_w^0, \hat{\mathcal{B}}_w^0$ with $w \in [3]$ and \mathcal{C}^0 satisfying $P_\Omega(\mathcal{C}^0 - \mathcal{M}) = 0$, $\Re(\mathcal{C}^0) = 0$, and the rank of $\hat{\mathcal{A}}^0, \hat{\mathcal{B}}^0$ are less than \mathbf{r} . Iterate the following steps for $t = 1, 2, \dots$ while it does not satisfy stop criterion.

Step 1. Compute

$$\mathcal{C}^t = \arg \min_{P_\Omega(\mathcal{C} - \mathcal{M})=0, \Re(\mathcal{C})=0} \sum_{w=1}^3 \frac{\alpha_w}{2} \|\mathcal{A}_w^{t-1} *_{\mu}^w \mathcal{B}_w^{t-1} - \mathcal{C}\|_F^2 + \sum_{k=1}^2 \lambda_k \|\mathcal{C} \times_k H_{n_k}\|_F^2 + \sum_{\alpha=\mathbf{i}, \mathbf{j}, \mathbf{k}} \langle \delta_\alpha^t, \mathcal{C}_\alpha \rangle$$

via Algorithm 3. That is, given parameters of Algorithm 3, apply PGM to find an approximate solution \mathcal{C}^t of (60) such that the error vector δ^t satisfies $\Re(\delta^t) = 0$, $P_\Omega(\delta^t) = 0$ and the accuracy condition

$$\|P_{\Omega^c}(\delta^t)\|_F \leq \frac{1}{4} \|\mathcal{C}^t - \mathcal{C}^{t-1}\|_F.$$

Step 2. Compute $\hat{\mathcal{A}}_w^t$ by (61).

Step 3. Compute $\hat{\mathcal{B}}_w^t$ by (62).

Step 5. Check the stop criterion:

$$\frac{|g(\mathcal{C}^k, \mathcal{A}_1^k, \mathcal{B}_1^k, \mathcal{A}_2^k, \mathcal{B}_2^k, \mathcal{A}_3^k, \mathcal{B}_3^k) - g(\mathcal{C}^{k-1}, \mathcal{A}_1^{k-1}, \mathcal{B}_1^{k-1}, \mathcal{A}_2^{k-1}, \mathcal{B}_2^{k-1}, \mathcal{A}_3^{k-1}, \mathcal{B}_3^{k-1})|}{|g(\mathcal{C}^{k-1}, \mathcal{A}_1^{k-1}, \mathcal{B}_1^{k-1}, \mathcal{A}_2^{k-1}, \mathcal{B}_2^{k-1}, \mathcal{A}_3^{k-1}, \mathcal{B}_3^{k-1})|} < \epsilon.$$

Output. \mathcal{C}^t .

frames. $\mathcal{C}(:, :, :, 1)$, $\mathcal{C}(:, :, :, 2)$ and $\mathcal{C}(:, :, :, 3)$ correspond to red, green and blue channels, respectively. The index set is Ω and the sampling ratio ρ is defined by

$$\rho = \frac{\text{numel}(\Omega)}{n_1 \times n_2 \times n_3}.$$

For TCTF, TNN and TMac3D, the three third-order real tensors for red, green and blue channels are first recovered. Then, the three recovered tensors are combined to form the integrated color video data. For TMac4D, we arrange the color video data as a fourth-order tensor and directly recover the incomplete part $P_\Omega(\mathcal{C})$. Totally there are four TMac-type methods, i.e., TMac3D-dec, TMac3D-inc, TMac4D-dec and TMac4D-inc. For LRQA-2 method, we recover each frame of video by LRQA-2 and finally combine them into an integrated video tensor. In our two methods, each color video is reshaped as a pure quaternion tensor $\mathcal{C} \in \mathbb{H}^{n_1 \times n_2 \times n_3}$ by using the following way:

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

All the simulations are run in MATLAB 2020b under Windows 10 on a laptop with 1.30 GHz CPU and 16GB memory.

5.1 Quantitative assessment and parameter settings

In order to evaluate the performance of QRTC and MQRTC, 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:

$$\text{RSE} = 10 \log 10 \left(\frac{\|\mathcal{C} - \hat{\mathcal{C}}\|_F}{\|\mathcal{C}\|_F} \right),$$

where $\hat{\mathcal{C}}$ and \mathcal{C} are the recovered and truth data, respectively.

$$\text{PSNR} = 10 \log 10 \left(\frac{\text{numel}(\hat{\mathcal{C}}) \text{Peakval}^2}{\|\hat{\mathcal{C}} - \mathcal{C}\|_F^2} \right),$$

where Peakval is taken from the range of the image datatype (e.g., for uint8 image, it is 255).

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

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

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

where Δ demotes the whole video spatial and temporal domain. The phase congruency for position x of video \mathcal{C} is denoted as $PC_{\mathcal{C}(x)}$, then $PC_m(x) = \max\{PC_{\mathcal{C}(x)}, PC_{\hat{\mathcal{C}}(x)}\}$, $S_L(x)$ is the gradient magnitude for position x .

Without special instructions, in the all experiments in Section 5, we set the initialized rank $\mathbf{r}^0 = [30, 30, 30]$ in TMac3D-dec, $\mathbf{r}^0 = [30, 30, 30, 30]$ in TMac4D-dec, $\mathbf{r}^0 = [3, 3, 3]$ in TMac3D-inc and $\mathbf{r}^0 = [3, 3, 3, 3]$ in TMac4D-inc, and set the weights for both versions as suggested in [19]. For TCTF, we set the initialized rank $\mathbf{r}^0 = [30, \dots, 30] \in \mathbb{R}^{n_3}$ the same as that in [3]. Following [34], we use LRQA with Laplace function penalty, and set parameter $\gamma = 20$. For all the methods except ours, the stopping criteria are built-in their codes. In our methods, the initial rank $\mathbf{r}^0 = [30, \dots, 30] \in \mathbb{R}^{n_3}$. An then we will give the same setting of weight parameter $\alpha_1, \alpha_2, \alpha_3$, the penalty parameter λ and TV penalty parameter λ_1, λ_2 in both QRTC and MQRTC. Noticing that two spatial dimension are symmetry, so we can naturally set $\alpha_1 = \alpha_2$ and $\lambda_1 = \lambda_2$. We Set α_3 as cardinality 1, and then as a matter of experience, we set $\alpha_1 = \alpha_2 = 10$, $\lambda_1 = \lambda_2 = 5$ and $\lambda = \alpha_1 + \alpha_2 + \alpha_3$. In experiments, the maximum iteration number is set to be 20 and the termination precision ϵ is set to be $1e-3$.

5.2 Performances of methods based on Qt-SVD and t-SVD

Both t-SVD [10] and our novel factorization Qt-SVD depict the inherent low rank structure of a third order real or quaternion tensor. Here we conduct experiments to compare them in detail on real color video data. Other methods are not compared here since they are not based on matrix factorization of a Fourier transform result. In order to show that Qt-SVD explores the low rank property better of color video, we fairly compare TCTF and our method in the similar formulation. Notice that the model of TCTF is given as (6), we set parameters $\lambda = \lambda_1 = \lambda_2 = 0$ in QRTC (28) to get the similar formulation **QRTC-1**, i.e.

$$\min_{\mathcal{C}, \mathcal{A}, \mathcal{B}} \frac{1}{2} \|\mathcal{A} *_{\mu} \mathcal{B} - \mathcal{C}\|_F^2, \text{ s.t. } P_{\Omega}(\mathcal{C} - \mathcal{M}) = 0, \Re(\mathcal{C}) = 0. \quad (63)$$

We test TCTF and QRTC-1 on fifteen real color videos data of YUV Video Sequences. The frame size of each video is 288×352 , and only the first 30 frames of each video are extracted as experimental data due to the computational limitation. The initialized rank of TCTF and QRTC-1 are set as $\mathbf{r}^0 = [30, \dots, 30] \in \mathbb{R}^{n_3}$.

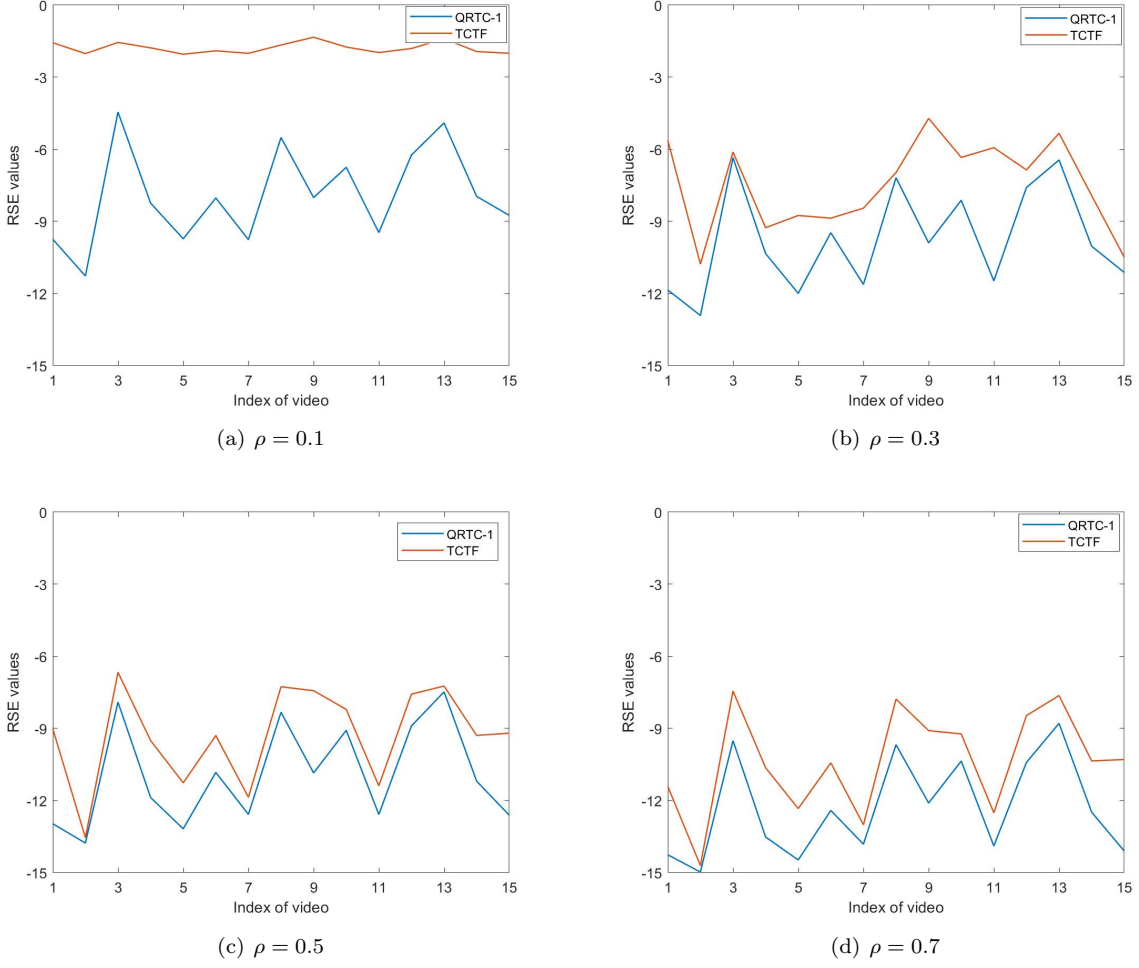


Figure 3: Comparison of RSE result of TCTF and QRTC-1 for color video recovery on 15 videos, sample ratio $\rho = 0.1, 0.3, 0.5$ and 0.7 . From RSE, our Qt-SVD based QRTC outperforms over t-SVD based TCTF. Moreover, the smaller of sample size, the performance of QRTC-1 is better from (a).

As shown in Figure 3, we display the RSE values of the recovery of fifteen video data with four sample ratios, $\rho = 0.1, 0.3, 0.5$ and 0.7 , respectively. We can see that the RSE values of our Qt-SVD based QRTC-1 are always less than of TCTF with all sample ratios. This shows the better performance and robustness of methods based on Qt-SVD. And when sample ratio $\rho = 0.1$, our method has a greater RSE, which shows QRTC-1 has better performance.

Table 1 displays the PSNR values of recovery of fifteen videos data with four sample ratios, $\rho = 0.1, 0.3, 0.5$ and 0.7 , respectively. The bold values in Table 1 is the best values of two methods. It is shown in Table 1 that our method always achieves the best. Especially with $\rho = 0.1$, the average PSNR values of our method is two times better than of TCTF, this shows our method has a greater development on exploring the low rank structure of color video data. Table 2 shows the average running time of two methods. We can see that the running time of QRTC-1 not longer than an order of magnitude with TCTF, which is an acceptable cost in practice.

5.3 Video inpainting for different methods

We first evaluate our method on the *videoSegmentationData* dataset, which can be downloaded in [41]. We test all the above mentioned methods on color video datasets “AN119T”, “BR128T”, “DO01-013”, “DO01-030” and “M07-058”. The frame size of all videos is 288×352 . We set the sampling ratio $\rho = 0.3$ and uniformly sample the videos to construct the observable index set Ω . All parameters are set

Index	$\rho = 0.1$		$\rho = 0.3$		$\rho = 0.5$		$\rho = 0.7$	
	TCTF	QRTC-1	TCTF	QRTC-1	TCTF	QRTC-1	TCTF	QRTC-1
Akiyo	10.0340	26.3871	18.1500	30.6066	24.9302	32.8357	29.7239	35.4168
Bridge(Close)	6.6788	25.1930	24.1929	28.4788	29.7362	30.1917	32.0634	32.6023
Bus	11.6938	17.4861	20.8127	21.2821	21.9240	24.3991	23.4783	27.6053
Coastguard	9.2534	22.1785	24.2265	26.3815	24.7123	29.4515	26.9737	32.7388
Container	8.1161	23.4795	21.5396	28.0197	26.5626	30.3911	28.7022	32.9819
Flower	5.3197	17.5676	19.2486	20.4618	20.1102	23.1790	22.3841	26.3466
Hall Monitor	8.6508	24.1589	21.5368	27.8881	28.3716	29.7974	30.6626	32.2779
Mobile	7.5761	15.2794	18.1969	18.6211	18.7906	20.9175	19.8210	23.6180
News	11.1349	24.4833	17.8918	28.2573	23.3263	30.1733	26.6495	32.6758
Paris	10.1639	20.1727	19.3490	22.9161	23.0965	24.8344	25.1367	27.3943
Silent	9.3471	24.3320	17.2662	28.3484	28.1783	30.5686	30.4278	33.1890
Stefan	8.6749	17.5420	18.7819	20.2443	20.2135	22.8605	21.9987	25.8947
Tempete	12.3150	19.3154	20.1789	22.4004	23.9860	24.4744	24.7784	27.0957
Waterfall	11.9192	23.9763	23.9104	28.1241	26.6389	30.4502	28.7643	30.0320
Foreman	7.1351	20.6288	24.1126	25.3807	21.5267	28.3540	23.7150	31.3344
Average	9.2009	21.4787	20.6263	25.1607	24.1403	27.5252	26.3520	30.2802

Table 1: Comparison of PSNR result of TCTF and QRTC-1 for color video recovery on 15 videos, sample ratio $\rho = 0.1, 0.3, 0.5, 0.7$.

$\rho = 0.1$		$\rho = 0.3$		$\rho = 0.5$		$\rho = 0.7$	
TCTF	QRTC-1	TCTF	QRTC-1	TCTF	QRTC-1	TCTF	QRTC-1
206	219	166	214	169	226	168	240

Table 2: Average running time (seconds) of TCTF and QRTC-1 for color video recovery on 15 videos, sample ratio $\rho = 0.1, 0.3, 0.5, 0.7$.

as mentioned above. The first frames of five examples of selected videos are shown in Figure 4. From Figure 4, it is seen that in all videos our two methods have better recovery of details on the marginal area between the main target and the surrounding environment. Table 3 summaries the RSE, PSNR, SSIM, FSIM values and the running time of all the algorithms on the five testing videos displayed in Figure 4. In Table 3, the bold values and the values in brackets stands for the best and the second best values of items RSE, PSNR, SSIM and FSIM, respectively. From the results, it is shown that the overall performance of QRTC and MQRTC are vastly superior to the others: the best of the above four quality assessments is consistently of QRTC or MQRTC, while MQRTC behaves better than MARTC in most of cases. The running time of QRTC is longer than several methods of TMac, but not longer than an order of magnitude. The running time of MQRTC is approximately fourfold as long as that of QRTC, but no more than that of TNN. All these outcomes demonstrate that in terms of color video inpainting problems, our methods have better recovery accuracy than others and runs also very efficiently.

To verify the robustness of our methods to the sampling ratio ρ , we test the video “Stefan” which is of YUV Video Sequences. The frame size of the video is 288×352 . We set the sampling ratio ρ ranging from 0.1 to 0.5 and uniformly sample the pixels to construct Ω . All the other parameters are set as mentioned. The first frame of the selected video with different sampling ratios are shown in Figure 5. Figure 5 indicates that the recovered videos of our methods are the clearest under all sampling ratios. Table 4 summaries the RSE, PSNR, SSIM, FSIM values and the running time of all the algorithms on the selected video with all sampling ratios which are displayed in Figure 5. In Table 4, the bold values and the values in brackets are the best and the second best values of RSE, PSNR, SSIM and FSIM, respectively. From the results, the best and the second best of PSNR, RSE or SSIM are of either QRTC or MQRTC. For FSIM value, MQRTC and QRTC performs better than others except the situation when $\rho = 0.1$. The running time of QRTC is longer than several methods of TMac, but not longer than an order of magnitude. The running time of MQRTC is approximately fourfold as long as that of QRTC, but not exceeds that of TNN or LRQA-2. Thus, we can conclude that our methods are rather robust to the sampling ratio and have



Figure 4: First frame of color video recovery using different algorithms ($\rho = 0.3$).

Table 3: Quantitative quality indexes and running time (seconds) of different algorithm on the five videos displayed in Figure 4 ($\rho = 0.3$).

Videos	Indexes	TCTF	TNN	TMac3D-dec	TMac3D-inc	TMac4D-dec	TMac4D-inc	LRQA-2	QRTC	MQRTC
AN119T	RSE	-10.1719	-11.3749	-10.5061	-10.9905	-10.6873	-11.1481	-10.3217	(-12.5983)	-12.7885
	PSNR	26.7931	29.1990	27.4615	28.4303	27.8239	28.7456	27.0927	(31.6458)	32.0263
	SSIM	0.8681	0.8984	0.8739	0.8864	0.8814	0.8922	0.8748	0.9681	(0.9498)
	FSIM	0.8687	0.8978	0.7984	0.8288	0.8170	0.8428	0.8689	(0.9175)	0.9249
	time(s)	241	2182	49	141	103	185	2460	173	744
BR128T	RSE	-8.2424	-9.7408	-7.5778	-8.4865	-7.7757	-8.6600	-8.3008	(-11.0355)	-11.2164
	PSNR	22.3657	25.3624	21.0364	22.8538	21.4322	23.2008	22.4825	(27.9519)	28.3137
	SSIM	0.7691	0.8684	0.7612	0.8168	0.7836	0.8311	0.7840	0.9376	(0.9299)
	FSIM	0.8444	0.8917	0.8063	0.8438	0.8193	0.8519	0.8445	0.9443	(0.9436)
	time(s)	402	1813	60	180	174	256	3908	221	923
DO01-013	RSE	-7.6358	-13.0349	-11.5504	-12.1381	-11.7081	-12.2760	-11.5340	(-13.8539)	-14.4249
	PSNR	22.1400	32.9382	29.9691	31.1446	30.2846	31.4203	29.9363	(34.5761)	35.7182
	SSIM	0.8758	0.9752	0.9600	0.9667	0.9638	0.9696	0.9502	(0.9863)	0.9883
	FSIM	0.8575	0.9405	0.8553	0.8804	0.8679	0.8888	0.8983	(0.9462)	0.9582
	time(s)	310	3241	59	140	122	186	2262	169	700
DO01-030	RSE	-6.9388	-12.0369	-10.6581	-11.3983	-10.6586	-11.3957	-10.1583	(-12.7410)	-13.2912
	PSNR	19.4925	29.6887	26.9312	28.4115	26.9322	28.4062	25.9315	(31.0969)	32.1972
	SSIM	0.8509	0.9763	0.9535	0.9646	0.9535	0.9642	0.9431	(0.9837)	0.9862
	FSIM	0.8436	0.9175	0.8551	0.8786	0.8568	0.8798	0.9527	(0.9370)	0.9481
	time(s)	364	5770	59	161	152	231	3116	196	793
M07-058	RSE	-11.8070	-13.9020	-13.8192	-14.5612	-13.9278	-14.6486	-12.7418	(-15.4564)	-15.9781
	PSNR	27.1223	31.3122	31.1466	32.6305	31.3638	32.8055	28.9919	(34.4211)	35.4644
	SSIM	0.9572	0.9805	0.9787	0.9839	0.9806	0.9849	0.9691	(0.9910)	0.9925
	FSIM	0.7969	0.8766	0.9155	0.9255	0.9155	0.9278	0.9033	0.9754	(0.9584)
	time(s)	203	4206	47	131	80	170	1854	135	542

considerably better performance than others.

6 Conclusions

We introduce the concept of gQt-product for third-order quaternion tensor and then define QDFT. Based the newly-defined QDFT, we introduce gQt-SVD of third-order quaternion tensors. We define gQt-rank for third-order quaternion tensor via its gQt-SVD and show the existence of low gQt-rank optimal approximation. We also generalize these results from mode-3 (QRTC) to three modes (MQRTC) of third-order quaternion tensor, and obtain multi-gQt-rank. Numerical experiments indicate that third-order quaternion tensors generated by color videos in real life have an inherent low-rank property.

Therefore, we establish low-rank quaternion tensor completion models based on gQt-rank and multi-gQt-rank to recover color videos with partial data loss. Using TV-regularization to capture the spatial stability feature, we obtain our novel tensor recovery models for color video inpainting. We present two ALS algorithms (Algorithms 2 and 4) to solve our models. Their convergence is established (see Subsection 4.3). Extensive numerical experiments indicate that our approaches QRTC and MQRTC outperform some existing state-of-the-arts methods on various video datasets with different sample ratios, which also demonstrate the robustness of our methods. Especially, MQRTC outperforms QRTC.

Acknowledgement

We are very grateful to Dr. Yongyong Chen for sharing us with the code in [34]. Also many thanks go to the authors for their open-source codes of their studies. The author Liping Zhang was supported by the National Natural Science Foundation of China (Grant No. 12171271).

References

- [1] N. Sidiropoulos, R. Bro, and G. Giannakis. Parallel factor analysis in sensor array processing. *IEEE Transactions on Signal Processing*, 48(8):2377–2388, 2000.
- [2] M. Vasilescu and Terzopoulos D. *Multilinear analysis of image ensembles: TensorFaces*. Springer, 2002.
- [3] P. Zhou, C. Lu, Z. Lin, and C. Zhang. Tensor factorization for low-rank tensor completion. *IEEE Transactions on Image Processing*, 27(3):1152–1163, 2018.
- [4] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3842–3849, 2014.
- [5] Z. Zhang and S. Aeron. Exact tensor completion using t-SVD. *IEEE Transactions on Signal Processing*, 65(6):1511–1526, 2017.
- [6] B. Savas and L. Eldén. Handwritten digit classification using higher order singular value decomposition. *Pattern Recognition*, 40(3):993–1003, 2007.
- [7] J. Carroll and J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of “Eckart-Young” decomposition. *Psychometrika*, 35(3):283–319, 1970.
- [8] R. Harshman. Foundations of the parafac procedure: Model and conditions for an “explanatory” multi-mode factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.
- [9] L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.
- [10] M. Kilmer and C. Martin. Factorization strategies for third-order tensor. *Linear Algebra and its Applications*, 435:641–658, 2011.

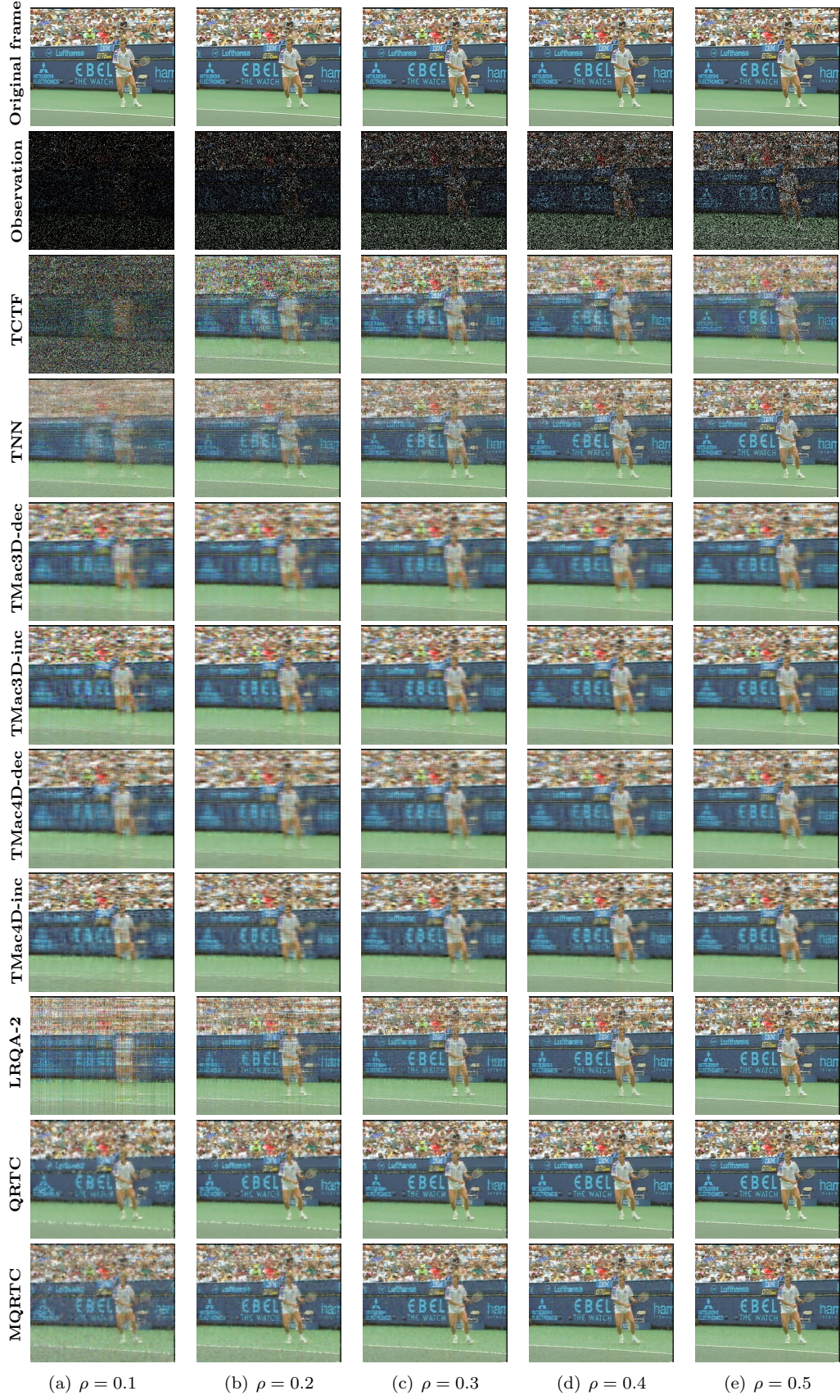


Figure 5: First frame of color video recovery while sample ratio ρ from 0.1 to 0.5 using different methods.

Table 4: Quantitative quality indexes and running time (seconds) of different algorithm on the “Stefan” video from $\rho = 0.1$ to 0.5 which are displayed in Figure 5.

ρ	Indexes	TCTF	TNN	TMac3D-dec	TMac3D-inc	TMac4D-dec	TMac4D-inc	LRQA-2	QRTC	MQRTC
$\rho = 0.1$	RSE	-1.79	-6.4133	-6.7203	-6.8304	-6.7406	-6.8705	-5.0637	(-6.9780)	-7.0966
	PSNR	8.6417	17.8723	18.4864	18.7066	18.5270	18.7869	15.1732	(19.0018)	19.2389
	SSIM	0.0818	0.6153	0.6620	0.6846	0.6687	0.6948	0.4232	0.7158	(0.7078)
	FSIM	0.6093	0.7068	0.6791	(0.7163)	0.6847	0.7190	0.6926	0.6905	0.7046
	time(s)	268	1201	141	183	328	331	3330	326	693
$\rho = 0.2$	RSE	-4.3148	-7.2013	-7.0963	-7.3819	-7.1831	-7.4486	-6.2521	(-7.9459)	-8.2292
	PSNR	13.6755	19.4483	19.2385	19.8096	19.4120	19.9429	17.5500	(20.9377)	21.5042
	SSIM	0.4807	0.7218	0.7091	0.7432	0.7240	0.7549	0.6079	(0.8156)	0.8256
	FSIM	0.7386	0.7787	0.6968	0.7374	0.7102	0.7460	0.7617	(0.7875)	0.8060
	time(s)	273	1241	70	138	131	185	3178	177	689
$\rho = 0.3$	RSE	-6.7642	-7.9653	-7.3641	-7.7258	-7.5270	-7.8525	-7.2777	(-8.7676)	-9.0605
	PSNR	18.5741	20.9765	19.7741	20.4974	20.0998	20.7508	19.6011	(22.5809)	23.1668
	SSIM	0.7206	0.7995	0.7416	0.7785	0.7643	0.7950	0.7327	(0.8752)	0.8815
	FSIM	0.8162	0.8310	0.7166	0.7578	0.7394	0.7736	0.8162	(0.8482)	0.8633
	time(s)	270	1319	48	123	117	170	2902	160	562
$\rho = 0.4$	RSE	-7.9171	-8.7597	-7.5969	-8.0123	-7.8549	-8.2227	-8.2366	(-9.5559)	-9.8308
	PSNR	20.8800	22.5653	20.2396	21.0704	20.7556	21.4911	21.5189	(24.1576)	24.7074
	SSIM	0.8125	0.8586	0.7677	0.8051	0.7977	0.8271	0.8206	(0.9149)	0.9174
	FSIM	0.8388	0.8728	0.7363	0.7775	0.7659	0.7987	0.8628	(0.8915)	0.9025
	time(s)	250	1448	42	115	104	160	2939	148	471
$\rho = 0.5$	RSE	-7.4885	-9.6432	-7.8107	-8.2800	-8.1682	-8.5860	-9.2639	(-10.3641)	-10.6105
	PSNR	20.0227	24.3323	20.6671	21.6058	21.3821	22.2178	23.5737	(25.7741)	26.2668
	SSIM	0.7902	0.9047	0.7891	0.8275	0.8257	0.8541	0.8805	0.94252	(0.94251)
	FSIM	0.8193	0.9076	0.7528	0.7942	0.7898	0.8215	0.9022	(0.9237)	0.9309
	time(s)	244	1746	39	120	85	146	2814	143	397

- [11] Liquan Qi, Yinnan Chen, Mayank Bakshi, and Xinzheng Zhang. Triple decomposition and tensor recovery of third order tensors. *SIAM Journal on Matrix Analysis and Applications*, 42(1):299–329, 2021.
- [12] N. Hao M. Kilmer, K. Braman and R. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34:148–172, 2013.
- [13] C. Hillar and L. Lim. Most tensor problems are NP-hard. *Journal of the ACM*, 60(6):1–39, 2013.
- [14] J. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.
- [15] 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.
- [16] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Math. Prog. Comp.*, 4:333–361, 2012.
- [17] E. Acar, D. Dunlavy, T. Kolda, and M. Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56, 2011.
- [18] L. Yang, Z.H. Huang, S. Hu, and J. Han. An iterative algorithm for third-order tensor multi-rank minimization. *Computational Optimization and Applications*, 63:169–202, 2016.
- [19] Y. Xu, R. Hao, W. Yin, and Z. Su. Parallel matrix factorization for low-rank tensor completion. *Inverse Problems and Imaging*, 9(2):601–624, 2013.
- [20] J. Liu, M. Przemyslaw, W. Peter, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.
- [21] B. Romera-Paredes and M. Pontil. A new convex relaxation for tensor completion. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, page 2967–2975. Curran Associates Inc., 2013.
- [22] O. Semerci, N. Hao, M. Kilmer, and E. Miller. Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. *IEEE Transactions on Image Processing*, 23(4):1678–1693, 2014.
- [23] T. Ell and S. Sangwine. Hypercomplex fourier transforms of color images. *IEEE Transactions on Image Processing*, 16(1):22–35, 2007.
- [24] Z. Jia, M. Ng, and G. Song. Robust quaternion matrix completion with applications to image inpainting. *Numerical Linear Algebra with Applications*, 26(4):e2245, 2019.
- [25] Z. Jia, M. Ng, and G. Song. Lanczos method for large-scale quaternion singular value decomposition. *Numerical Algorithms*, 82:699–717, 2019.
- [26] Z. Subakan and B. Vemuri. A quaternion framework for color image smoothing and segmentation. *International Journal of Computer Vision*, 91(3):233–250, 2011.
- [27] S. Pei and C. Cheng. A novel block truncation coding of color images using a quaternion-moment-preserving principle. *IEEE Transactions on Communications*, 45(5):583–595, 1997.
- [28] Z. Qin, Z. Ming, and L. Zhang. Singular value decomposition of third order quaternion tensors. *Applied Mathematics Letters*, 123:107597, 2022.
- [29] F. Zhang. Quaternions and matrices of quaternions. *Linear Algebra and its Applications*, 251:21–57, 1997.
- [30] J. Nelson. Lecture note: Cs 229r: Algorithms for big data: Lecture 22. <http://people.seas.harvard.edu/~minilek/cs229r/fall115/lec/lec22.pdf>.

- [31] T. Ell. Quaternion-fourier transforms for analysis of two-dimensional linear time-invariant partial differential systems. In *Proceedings of 32nd IEEE Conference on Decision and Control*, pages 1830–1841, 1993.
- [32] G. Golub and C. Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.
- [33] Q. Yu, X. Zhang, and Z. Huang. Multi-tubal rank of third order tensor and related low rank tensor completion problem. arXiv:2012.05065v1.
- [34] Y. Chen, X. Xiao, and Y. Zhou. Low-rank quaternion approximation for color image processing. *IEEE Transactions on Image Processing*, 29:1426–1439, 2020.
- [35] G. Song, W. Ding, and M. Ng. Robust quaternion matrix completion with applications to image inpainting. *SIAM Journal on Matrix Analysis and Application*, 42(1):58–82, 22021.
- [36] J. Barzilai and J. Borwein. Two point step-size gradient methods. *Ima Journal of Numerical Analysis*, 8:141–148, 1988.
- [37] Y. Chen, L. Qi, X. Zhang, and Y. Xu. A low rank quaternion decomposition algorithm and its application in color image inpainting. arXiv:2009.12203.
- [38] J. Bolte H. Attouch and B. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized gauss-seidel methods. *Mathematical Programming*, 137:91–129, 2013.
- [39] H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116:5–16, 2009.
- [40] J. Bolte, A. Daniilidis, and A. Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, November 2006.
- [41] K. Fukuchi, K. Miyazato, A. Kimura, S. Takagi, and J. Yamato. Saliency-based video segmentation with graph cuts and sequentially updated priors. In *2009 IEEE International Conference on Multimedia and Expo*, pages 638–641, 2009.