

Efficient quaternion CUR method for low-rank approximation to quaternion matrix

Pengling Wu

University of Macau

Kit Ian Kou

kikou@um.edu.mo

University of Macau

Hongmin Cai

South China University of Technology

Zhaoyuan Yu

Nanjing Normal University

Research Article

Keywords: Quaternion matrix, quaternion CUR decomposition, low-rank approximation, color image processing

Posted Date: March 1st, 2024

DOI: <https://doi.org/10.21203/rs.3.rs-3989824/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Numerical Algorithms on August 22nd, 2024. See the published version at <https://doi.org/10.1007/s11075-024-01923-8>.

Efficient quaternion CUR method for low-rank approximation to quaternion matrix

Pengling Wu¹, Kit Ian Kou^{1*}, Hongmin Cai², Zhaoyuan Yu³

^{1*}Department of Mathematics, Faculty of Science and Technology, University of Macau, Macau 100190, China.

²School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China.

³Department, School of Geography, Nanjing Normal University, Nanjing 210023, China.

*Corresponding author(s). E-mail(s): kikou@um.edu.mo;
Contributing authors: wupengling@163.com; hmcai@scut.edu.cn;
yuzhaoyuan@nynu.edu.cn;

Abstract

The low-rank quaternion matrix approximation has been successfully applied in many applications involving signal processing and color image processing. However, the cost of quaternion models for generating low-rank quaternion matrix approximation is sometimes considerable due to the computation of the quaternion singular value decomposition (QSVD), which limits their application to real large-scale data. To address this deficiency, an efficient quaternion matrix CUR (QMCUR) method for low-rank approximation is suggested, which provides significant acceleration in color image processing. We first explore the QMCUR approximation method, which uses actual columns and rows of the given quaternion matrix, instead of the costly QSVD. Additionally, two different sampling strategies are used to sample the above-selected columns and rows. Then, the perturbation analysis is performed on the QMCUR approximation of noisy versions of low-rank quaternion matrices. Extensive experiments on both synthetic and real data further reveal the superiority of the proposed algorithm compared with other algorithms for getting low-rank approximation, in terms of both efficiency and accuracy.

Keywords: Quaternion matrix, quaternion CUR decomposition, low-rank approximation, color image processing

1 Introduction

Quaternion [1] as a mathematical concept was originally introduced by Hamilton in 1843. As an extension of complex numbers, a quaternion number consists of one real part and three imaginary parts. A quaternion matrix is a generalization of a complex matrix in quaternion algebra. By now, quaternions and quaternion matrices have a wide range of applications in signal processing [2–4], machine learning [5, 6], and particularly in color image processing [7–11]. By encoding the red, green, and blue channel pixel values of a color image on the three imaginary parts of quaternion matrices, this method perfectly fits the color image structure and effectively preserves the inter-relationship between the color channels [12].

As an emerging mathematical tool, low-rank quaternion matrix approximation (LRQA) has attracted much attention in the field of color image processing, such as color face recognition [13, 14], color image inpainting [15–17], and color image denoising [18, 19]. A variety of LRQA-based variants are achieved by using different rank approximation regularizers [20–22]. Notably, Chen *et al.* [23] proposed the quaternion nuclear norm (QNN)-based LRQA for color image denoising and inpainting. This method fully utilizes the high correlation among RGB channels by extending low-rank matrix approximation into the quaternion domain. Yu *et al.* [24] further extended the weighted nuclear norm minimization (WNNM) into the quaternion domain and proposed the quaternion-based WNNM method for color image restoration. Additionally, in [25] and [26], the authors employed quaternion truncated nuclear norm and logarithmic norm to achieve a more accurate low-rank approximation. However, a key limitation of these methods for generating low-rank quaternion matrix approximation is that they need to compute the quaternion singular value decompositions (QSVD) in each iteration and suffer from computational deficiency, especially for large-scale data.

In addition to rank minimization, recent studies have utilized quaternion matrix decomposition and randomized techniques to improve the performance of LRQA [17, 27, 28]. For instance, Miao *et al.* [29] suggested the matrix factorization for the target quaternion matrix, followed by three quaternion-based bilinear factor matrix norm factorization methods for low-rank quaternion matrix completion, which can avoid expensive calculations. Liu *et al.* [30] presented a randomized QSVD algorithm for low-rank matrix approximation. The randomized QSVD algorithm reduces the computational cost compared to traditional QSVD for large-scale data. Is there any method with lower computational cost for low-rank approximation of a quaternion matrix?

In recent years, the matrix CUR (MCUR) method [31–33] for fast low-rank approximation of real matrices has been actively investigated because of its ability to efficiently handle large-scale problems. The MCUR method approximates a low-rank matrix by directly sampling a subset of columns and rows from the original matrix and representing it as a product of three small-scale matrices. Owing to the random column/row selection strategy, the MCUR decomposition shows great potential for reducing the computational costs and preserving the properties of the original data matrix compared with other methods [34, 35].

To further enhance the approximation performance and boost the computational efficiency of LRQA, we consider the efficient quaternion matrix CUR (QMCUR) method for low-rank approximation. Precisely, the QMCUR approximation of a quaternion matrix is obtained by utilizing the actual rows and columns of the original quaternion matrix. Additionally, we employ two different sampling strategies to sample the above-selected columns and rows. In summary, the introduction of the QMCUR approximation and the utilization of

sampling strategies offer promising avenues for achieving low-rank approximation. The main contributions of this paper are twofold:

- We consider the QMCUR method for low-rank approximation of a quaternion matrix. This approach helps to reduce computational costs and improve the precision of low-rank approximation.
- The perturbation error bound of the proposed approximation method is studied and demonstrated under the spectral norm. Experimental results for synthetic data and color images demonstrate that the proposed QMCUR approximation method achieves a substantial speed-up while maintaining accuracy compared with other methods.

The rest of the paper is organized as follows. The main notation and preliminary for quaternions are presented in Section 2. Section 3 presents the proposed quaternion CUR approximation method and analyzes its perturbation error bound. We provide numerical experiments to illustrate the effectiveness of the proposed method in Section 5. Finally, some concluding remarks will be given in Section 6.

2 Preliminaries

2.1 Notation

Throughout this paper, $\mathbb{R}$ and $\mathbb{H}$ respectively denote the real and quaternion spaces. The set of all positive integers is denoted by $\mathbb{N}$, and the symbol $[n]$ represents the set of integers $\{1, \dots, n\}$ for any $n \in \mathbb{N}$. A scalar, vector, and matrix are written as a , $\mathbf{a}$, and $\mathbf{A}$, respectively. We use $\mathbf{A}(I, :)$ and $\mathbf{A}(:, J)$ to denote the row and column submatrices with indices I and J , respectively. Here, $(\cdot)^*$, $(\cdot)^T$, $(\cdot)^H$, and $(\cdot)^\dagger$ represent the conjugate, transpose, conjugate transpose, and Moore-Penrose inverse, respectively. $|\cdot|$ and $\|\cdot\|_F$ represent the absolute value or modulus and the Frobenius norm, respectively.

2.2 Quaternion and quaternion matrices

The set of quaternions $\mathbb{H}$ is a linear space over $\mathbb{R}$, with an ordered basis of 1, $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$. Here, $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ are three imaginary units with the multiplication laws: $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$, $\mathbf{ij} = -\mathbf{ji} = \mathbf{k}$, $\mathbf{jk} = -\mathbf{kj} = \mathbf{i}$, $\mathbf{ki} = -\mathbf{ik} = \mathbf{j}$. An element a of the set of quaternions $\mathbb{H}$ is of the form $a = a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k} \in \mathbb{H}$ with $a_0, a_1, a_2, a_3 \in \mathbb{R}$. In particular, a is called a pure quaternion if the real component a_0 equals 0. The conjugate and modulus of a are defined as $a^* = a_0 - a_1\mathbf{i} - a_2\mathbf{j} - a_3\mathbf{k}$ and $|a| = \sqrt{a_0^2 + a_1^2 + a_2^2 + a_3^2}$, respectively.

For an arbitrary quaternion matrix $\mathbf{A} = \mathbf{A}_0 + \mathbf{A}_1\mathbf{i} + \mathbf{A}_2\mathbf{j} + \mathbf{A}_3\mathbf{k} = (a_{st}) \in \mathbb{H}^{m \times n}$ with $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3 \in \mathbb{R}^{m \times n}$, $\mathbf{A}^H = (a_{ts}^*) \in \mathbb{H}^{n \times m}$ denotes the conjugate transpose of $\mathbf{A}$. Moreover, the Frobenius norm of a quaternion matrix $\mathbf{A} \in \mathbb{H}^{m \times n}$ is $\|\mathbf{A}\|_F = \sqrt{\text{Tr}(\mathbf{A}^H \mathbf{A})} = \sqrt{\sum_{s=1}^m \sum_{t=1}^n |a_{st}|^2}$, where $\text{Tr}(\cdot)$ is the trace operator, and the spectral norm $\|\mathbf{A}\|_2$ is defined as $\|\mathbf{A}\|_2 = \max_{\mathbf{x} \in \mathbb{H}^n, \|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2$.

The quaternion singular value decomposition (QSVD) [36] of a quaternion matrix $\mathbf{A} \in \mathbb{H}^{m \times n}$ is defined as $\mathbf{A} = \mathbf{W}\mathbf{\Sigma}\mathbf{V}^H$, where $\mathbf{W} \in \mathbb{H}^{m \times m}$ and $\mathbf{V} \in \mathbb{H}^{n \times n}$ are unitary, and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_l) \in \mathbb{R}^{m \times n}$ with $\sigma_i \geq 0$ denoting the i -th largest singular value of $\mathbf{A}$ and $l = \min\{m, n\}$. The truncated QSVD of a quaternion matrix $\mathbf{A}$ with rank k will be denoted by $\mathbf{A}_k = \mathbf{W}_k \mathbf{\Sigma}_k \mathbf{V}_k^H$, where $\mathbf{W} \in \mathbb{H}^{m \times k}$ and $\mathbf{V} \in \mathbb{H}^{n \times k}$ are the $m \times k$ and $n \times k$ submatrices of $\mathbf{W}$ and $\mathbf{V}$, respectively, corresponding to choosing the first k columns of each,

and Σ_k is a $k \times k$ matrix containing the largest k singular values. More information about quaternion matrices can be found in [12, 36, 37].

3 Proposed approach

In this section, we first present the QMCUR method, which is designed to efficiently compute a low-rank approximation of a large-scale quaternion matrix with low computational costs and comparable accuracy. Then we provide the perturbation estimates for the QMCUR approximation of the noisy version of low-rank matrices.

Algorithm 1 Main steps of QMCUR approximation

Input: $\mathbf{X} \in \mathbb{H}^{m \times n}$: the quaternion matrix; k : predefined rank; $|I|, |J|$: number of rows and columns to sample; $\{p_j\}, \{q_i\}$: sampling probability distribution, $j = 1, \dots, n$, and $i = 1, \dots, m$.

- 1: Draw sampling column indices $J \subseteq [n]$ based on the sampling probability p_j and construct $\mathbf{C} \in \mathbb{H}^{m \times |J|}$;
- 2: Draw sampling row indices $I \subseteq [m]$ based on the sampling probability q_i and construct $\mathbf{R} \in \mathbb{H}^{|I| \times n}$;
- 3: Compute $\mathbf{U}$: $\mathbf{U} = \mathbf{C}^\dagger \mathbf{X} \mathbf{R}^\dagger \in \mathbb{H}^{|I| \times |J|}$;

Output: $\mathbf{C}, \mathbf{U}, \mathbf{R}$ such that $\mathbf{X} \approx \mathbf{C} \mathbf{U} \mathbf{R}$.

3.1 QMCUR-based low-rank approximation method

We now present the QMCUR approximation method. Let $\mathbf{X} \in \mathbb{H}^{m \times n}$ be a low-rank quaternion matrix with a predefined rank k . Consider row indices $I \subseteq [m]$ and column indices $J \subseteq [n]$ satisfying $|I|, |J| \geq k$. Denote a column submatrix $\mathbf{C} = \mathbf{X}(:, J)$ whose columns span the column space of $\mathbf{X}$ and a row submatrix $\mathbf{R} = \mathbf{X}(I, :)$ whose rows span the row space of $\mathbf{X}$. Then the QMCUR approximation of $\mathbf{X}$ is a product of the form

$$\mathbf{X} \approx \mathbf{C} \mathbf{U} \mathbf{R}, \quad (1)$$

where $\mathbf{C} \in \mathbb{H}^{m \times |J|}$ with $|J|$ columns from the quaternion matrix $\mathbf{X}$ and $\mathbf{R} \in \mathbb{H}^{|I| \times n}$ with $|I|$ rows from the quaternion matrix $\mathbf{X}$. It's obvious that the core quaternion matrix $\mathbf{U} \in \mathbb{H}^{|J| \times |I|}$ should be computed to yield the smallest error. The optimal choice for the core quaternion matrix $\mathbf{U}$ in the least-squares sense is $\mathbf{U} = \mathbf{C}^\dagger \mathbf{X} \mathbf{R}^\dagger$ [38] because

$$\mathbf{C}^\dagger \mathbf{X} \mathbf{R}^\dagger = \arg \min_{\mathbf{U} \in \mathbb{H}^{|J| \times |I|}} \|\mathbf{X} - \mathbf{C} \mathbf{U} \mathbf{R}\|_F^2. \quad (2)$$

Note that one may replace $\mathbf{C}^\dagger$ in (1) by $\mathbf{C}^\dagger \mathbf{X} \mathbf{R}^\dagger$ to improve the approximation, resulting in the QMCUR approximation $\mathbf{C} \mathbf{C}^\dagger \mathbf{X} \mathbf{R}^\dagger \mathbf{R}$. Algorithm 1 summarizes the overall process of the QMCUR approximation.

To implement the QMCUR approximation method, we set the size of $\mathbf{C}$ to be $m \times k \log k$ and $\mathbf{R}$ to be $k \log k \times n$ for the QMCUR approximation method. The indices used to determine

$\mathbf{C}$ and $\mathbf{R}$ are sampled using two different strategies [39, 40]. In the first strategy, the sampling probabilities p_j^{col} and q_i^{row} for each column j and row i of the quaternion matrix $\mathbf{X}$ are based on the Euclidean norm of the columns and rows, which are respectively defined as $p_j^{\text{col}} := \frac{\|\mathbf{X}(:,j)\|_2^2}{\|\mathbf{X}\|_F^2}$ where $j = 1, 2, \dots, n$ and $q_i^{\text{row}} := \frac{\|\mathbf{X}(i,:)\|_2^2}{\|\mathbf{X}\|_F^2}$, where $i = 1, 2, \dots, m$. This strategy is referred to as QMCUR_length. In the second strategy, the sampling probabilities p_j^{unif} and q_i^{unif} are respectively defined as $p_j^{\text{unif}} := \frac{1}{n}$ and $q_i^{\text{unif}} := \frac{1}{m}$. This approach is referred to as QMCUR_uniform.

3.2 Perturbation Estimates for CUR Approximation

In practical applications, quaternion matrix data is often perturbed, such as noise in pictures, which may cause huge errors. In this section, we present the perturbation analysis suggested by the QMCUR approximation described above. Our main task will be to consider matrices in the form of

$$\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{E}, \quad (3)$$

where $\mathbf{X} \in \mathbb{H}^{m \times n}$ has rank k with $k \geq \{m, n\}$, and $\mathbf{E} \in \mathbb{H}^{m \times n}$ is an arbitrary noise quaternion matrix drawn from a certain distribution.

Before analyzing the perturbation, we will first introduce some notation. Let $\tilde{\mathbf{C}} = \tilde{\mathbf{X}}(:, J) \in \mathbb{H}^{m \times |J|}$, $\tilde{\mathbf{R}} = \tilde{\mathbf{X}}(I, :) \in \mathbb{H}^{|I| \times n}$, and $\tilde{\mathbf{U}} = \tilde{\mathbf{C}}^\dagger \tilde{\mathbf{X}} \tilde{\mathbf{R}}^\dagger \in \mathbb{H}^{|J| \times |I|}$ for selected index sets I, J . Then we can express this as

$$\tilde{\mathbf{C}} = \mathbf{C} + \mathbf{E}_J, \tilde{\mathbf{R}} = \mathbf{R} + \mathbf{E}_I, \tilde{\mathbf{U}} = \mathbf{U} + \mathbf{E}, \quad (4)$$

where $\mathbf{C} = \mathbf{X}(:, J)$, $\mathbf{R} = \mathbf{X}(I, :)$, and $\mathbf{U} = \mathbf{C}^\dagger \mathbf{X} \mathbf{R}^\dagger$. For simplicity, we write $\mathbf{E}_J = \mathbf{E}(:, J)$ and $\mathbf{E}_I = \mathbf{E}(I, :)$. The following theorem provides a perturbation estimate for the QMCUR approximation.

Theorem 1 Let $\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{E}$ for a fixed but arbitrary $\mathbf{E} \in \mathbb{H}^{m \times n}$. Using the notation in (4), then the following holds:

$$\|\mathbf{X} - \tilde{\mathbf{C}} \tilde{\mathbf{C}}^\dagger \tilde{\mathbf{X}} \tilde{\mathbf{R}}^\dagger \tilde{\mathbf{R}}\| \leq \|\mathbf{E}_I\| \|\mathbf{X} \mathbf{R}^\dagger\| + \|\mathbf{E}_J\| \|\mathbf{C}^\dagger \mathbf{X}\| + 3\|\mathbf{E}\|. \quad (5)$$

Furthermore,

$$\|\mathbf{X} - \tilde{\mathbf{C}} \tilde{\mathbf{C}}^\dagger \tilde{\mathbf{X}} \tilde{\mathbf{R}}^\dagger \tilde{\mathbf{R}}\| \leq \|\mathbf{E}\| (\|\mathbf{W}_{k,I}^\dagger\| + \|\mathbf{V}_{k,J}^\dagger\| + 3). \quad (6)$$

where $\mathbf{W}_{k,I} = \mathbf{W}_k(I, :) \in \mathbb{H}^{|I| \times k}$ and $\mathbf{V}_{k,J} = \mathbf{V}_k(J, :) \in \mathbb{H}^{|J| \times k}$ for index sets I, J .

In a word, Theorem 3.1 shows that the error estimation in the QMCUR method is controlled by the pseudoinverses of the submatrices of the orthogonal singular vectors and is linear in the norm of the noise $\mathbf{E}$. To establish Theorem 3.1, we will introduce the following lemmas.

Lemma 1 Suppose that $\mathbf{X} \in \mathbb{H}^{m \times n}$ has rank k with the QMCUR approximation: $\mathbf{X} \approx \mathbf{C} \mathbf{U} \mathbf{R}$, where $\mathbf{C} = \mathbf{X}(:, J)$ with selected column indices J , $\mathbf{R} = \mathbf{X}(I, :)$ with selected row indices I , and $\mathbf{U} = \mathbf{C}^\dagger \mathbf{X} \mathbf{R}^\dagger$. Then we have

$$\text{rank}(\mathbf{C}) = \text{rank}(\mathbf{R}) = \text{rank}(\mathbf{X}). \quad (7)$$

Proof. Recall that $\mathbf{C} = \mathbf{X}(:, J)$. Thus, $\text{span}(\mathbf{C}) \subset \text{span}(\mathbf{X})$. On account of the fact that $\mathbf{X} = \mathbf{C}\mathbf{C}^\dagger\mathbf{X}\mathbf{R}^\dagger\mathbf{R}$, we have $\text{span}(\mathbf{X}) \subset \text{span}(\mathbf{C})$. A similar argument shows that $\text{span}(\mathbf{R}^H) \subset \text{span}(\mathbf{X}^H)$. The conclusion follows. $\square$

Lemma 2 The following hold:

$$\|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger\tilde{\mathbf{X}}\| \leq \|\mathbf{E}_J\| \|\mathbf{C}^\dagger\mathbf{X}\| + \|\mathbf{E}\|, \quad \|\mathbf{X} - \tilde{\mathbf{X}}\tilde{\mathbf{R}}^\dagger\tilde{\mathbf{R}}\| \leq \|\mathbf{E}_I\| \|\mathbf{X}\mathbf{R}^\dagger\| + \|\mathbf{E}\|. \quad (8)$$

Proof. Note that

$$\begin{aligned} \|(\mathbf{I} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger)\mathbf{C}\| &= \|(\mathbf{I} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger)\tilde{\mathbf{C}} - (\mathbf{I} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger)\mathbf{E}_J\| \\ &\leq \|(\mathbf{I} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger)\tilde{\mathbf{C}}\| + \|(\mathbf{I} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger)\mathbf{E}_J\| \\ &\leq \|\mathbf{E}_J\|. \end{aligned} \quad (9)$$

The final inequality holds because the first norm term is 0 by identity of the MoorePenrose pseudoinverse and $\|\mathbf{I} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger\|_2 \leq 1$ as this is an orthogonal projection operator. Since $\text{rank}(\mathbf{C}) = \text{rank}(\mathbf{X})$, we have $\mathbf{X} = \mathbf{C}\mathbf{C}^\dagger\mathbf{X}$. Then

$$\begin{aligned} \|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger\tilde{\mathbf{X}}\| &\leq \|(\mathbf{I} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger)\mathbf{X}\| + \|\mathbf{E}\| \\ &= \|(\mathbf{I} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger)\mathbf{C}\mathbf{C}^\dagger\mathbf{X}\| + \|\mathbf{E}\| \\ &\leq \|\mathbf{E}_J\| \|\mathbf{C}^\dagger\mathbf{X}\| + \|\mathbf{E}\|, \end{aligned} \quad (10)$$

The second inequality is derived by mimicking the above argument. The conclusion follows. $\square$

Lemma 3 Suppose that $\mathbf{X} \in \mathbb{H}^{m \times n}$ has rank k and its truncated QSVD is $\mathbf{X} = \mathbf{W}_k \mathbf{\Sigma}_k \mathbf{V}_k^H$. Let $\mathbf{C}$ and $\mathbf{R}$ be submatrices of $\mathbf{X}$ (with selected row and column indexes I and J , respectively) and $\mathbf{U} = \mathbf{C}^\dagger\mathbf{X}\mathbf{R}^\dagger$ such that $\mathbf{X} = \mathbf{C}\mathbf{U}\mathbf{R}$. Then the following hold:

$$\|\mathbf{X}\mathbf{R}^\dagger\| = \|\mathbf{W}_{k,I}^\dagger\|, \quad \|\mathbf{C}^\dagger\mathbf{X}\| = \|(\mathbf{V}_{k,J}^H)^\dagger\|. \quad (11)$$

where $\mathbf{W}_{k,I} = \mathbf{W}_k(I, :) \in \mathbb{H}^{|I| \times k}$ and $\mathbf{V}_{k,J} = \mathbf{V}_k(J, :) \in \mathbb{H}^{|J| \times k}$ for selected index sets I, J .

Proof. Begin with the fact that $\mathbf{R} = \mathbf{W}_k(I, :)\mathbf{\Sigma}_k\mathbf{V}_k^H = \mathbf{W}_{k,I}\mathbf{\Sigma}_k\mathbf{V}_k^H$. Then we have

$$\mathbf{A}\mathbf{R}^\dagger = \mathbf{W}_k\mathbf{\Sigma}_k\mathbf{V}_k^H(\mathbf{W}_{k,I}\mathbf{\Sigma}_k\mathbf{V}_k^H)^\dagger. \quad (12)$$

Notice that $\mathbf{W}_{k,I}$ has full column rank, $\mathbf{\Sigma}_k$ is a $k \times k$ matrix with full rank, and $\mathbf{V}_k^H$ has orthogonal rows (i.e., $(\mathbf{V}_k^H)^\dagger = \mathbf{V}_k$). In this case,

$$(\mathbf{W}_{k,I}\mathbf{\Sigma}_k\mathbf{V}_k^H)^\dagger = (\mathbf{V}_k^H)^\dagger\mathbf{\Sigma}_k^{-1}\mathbf{W}_{k,I}^\dagger = \mathbf{V}_k\mathbf{\Sigma}_k^{-1}\mathbf{W}_{k,I}^\dagger. \quad (13)$$

Consequently, the following holds:

$$\begin{aligned}
\|\mathbf{X}\mathbf{R}^\dagger\| &= \|\mathbf{W}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^H \mathbf{V}_k \boldsymbol{\Sigma}_k^{-1} \mathbf{W}_{k,I}^\dagger\| \\
&= \|\boldsymbol{\Sigma}_k \mathbf{V}_k^H \mathbf{V}_k \boldsymbol{\Sigma}_k^{-1} \mathbf{W}_{k,I}^\dagger\| \\
&= \|\boldsymbol{\Sigma}_k \boldsymbol{\Sigma}_k^{-1} \mathbf{W}_{k,I}^\dagger\| \\
&= \|\mathbf{W}_{k,I}^\dagger\|.
\end{aligned} \tag{14}$$

The second equality above follows from the unitary invariance of the norm [37]. Similarly, we have

$$\|\mathbf{C}^\dagger \mathbf{X}\| = \|(\mathbf{V}_{k,J}^H)^\dagger\|, \tag{15}$$

whereupon the conclusion follows from the fact that $\|(\mathbf{V}_{k,J}^H)^\dagger\| = (\mathbf{V}_{k,J}^\dagger)^H$ which has the same norm as $\mathbf{V}_{k,J}^\dagger$. The conclusion follows. $\square$

Proof of Theorem 1. First note that

$$\begin{aligned}
\|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger \tilde{\mathbf{X}}\tilde{\mathbf{R}}^\dagger \tilde{\mathbf{R}}\| &= \|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger \tilde{\mathbf{X}} + \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger \tilde{\mathbf{X}}(\mathbf{I} - \mathbf{R}^\dagger \mathbf{R})\| \\
&\leq \|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger \tilde{\mathbf{X}}\| + \|\tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger\| \|\tilde{\mathbf{X}}(\mathbf{I} - \mathbf{R}^\dagger \mathbf{R})\| \\
&\leq \|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger \tilde{\mathbf{X}}\| + \|\tilde{\mathbf{X}}(\mathbf{I} - \mathbf{R}^\dagger \mathbf{R})\|.
\end{aligned} \tag{16}$$

The second inequality follows from the fact that $\|\mathbf{C}\mathbf{C}^\dagger\|_F = 1$. Next, using the formula $\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{E}$, we have $\|\tilde{\mathbf{X}}(\mathbf{I} - \tilde{\mathbf{R}}^\dagger \tilde{\mathbf{R}})\| \leq \|\mathbf{X}(\mathbf{I} - \tilde{\mathbf{R}}^\dagger \tilde{\mathbf{R}})\| + \|\mathbf{E}\|$ since $\|\mathbf{I} - \tilde{\mathbf{R}}^\dagger \tilde{\mathbf{R}}\|_2 \leq 1$. Then applying Lemma 2, we obtain

$$\begin{aligned}
\|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger \tilde{\mathbf{X}}\tilde{\mathbf{R}}^\dagger \tilde{\mathbf{R}}\| &\leq \|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger \tilde{\mathbf{X}}\| + \|\mathbf{X}(\mathbf{I} - \tilde{\mathbf{R}}^\dagger \tilde{\mathbf{R}})\| + \|\mathbf{E}\| \\
&\leq \|\mathbf{E}_I\| \|\mathbf{X}\mathbf{R}^\dagger\| + \|\mathbf{E}_J\| \|\mathbf{C}^\dagger \mathbf{X}\| + 3\|\mathbf{E}\|.
\end{aligned} \tag{17}$$

Moreover, from Lemma 3, it follows that

$$\|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\dagger \tilde{\mathbf{X}}\tilde{\mathbf{R}}^\dagger \tilde{\mathbf{R}}\| \leq \|\mathbf{E}\|(\|\mathbf{W}_{k,I}^\dagger\| + \|\mathbf{V}_{k,J}^\dagger\| + 3). \tag{18}$$

4 Numerical experiments

In this section, we conduct numerical experiments to assess the accuracy and computation time of the proposed QMCUR method for computing low-rank approximation of a quaternion data matrix. Several real-world datasets are employed to demonstrate our proposed algorithm.

Example 4.1 In this experiment, we will demonstrate error estimation in Theorem 1 using simulation data. We randomly generate a quaternion matrix $\mathbf{X}$ as a product $\mathbf{X} = \mathbf{W}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^H \in \mathbb{H}^{m \times m}$ and set its rank k to be 50, where $\mathbf{W}_k \in \mathbb{H}^{m \times k}$, $\mathbf{V}_k \in \mathbb{H}^{m \times k}$ are unitary matrices, and $\boldsymbol{\Sigma}_k$ is a diagonal $k \times k$ matrix with positive diagonal elements. In addition, the random noise quaternion matrix is given by $\mathbf{E} = \mathbf{E}_0 + \mathbf{E}_1 \mathbf{i} + \mathbf{E}_2 \mathbf{j} + \mathbf{E}_3 \mathbf{k} \in \mathbb{H}^{m \times n}$, where the entries of $\mathbf{E}_t (t = 0, 1, 2, 3)$ are i.i.d. Gaussian variables with mean zero and variance σ . Then we obtain the perturbed quaternion matrix

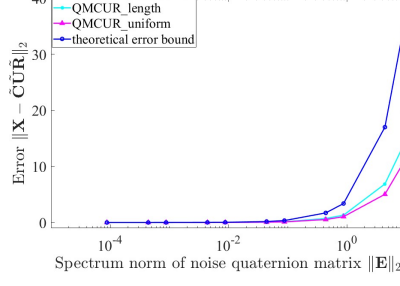


Fig. 1 $\|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{U}}\tilde{\mathbf{R}}\|_2$ vs. $\|\mathbf{E}\|_2$.

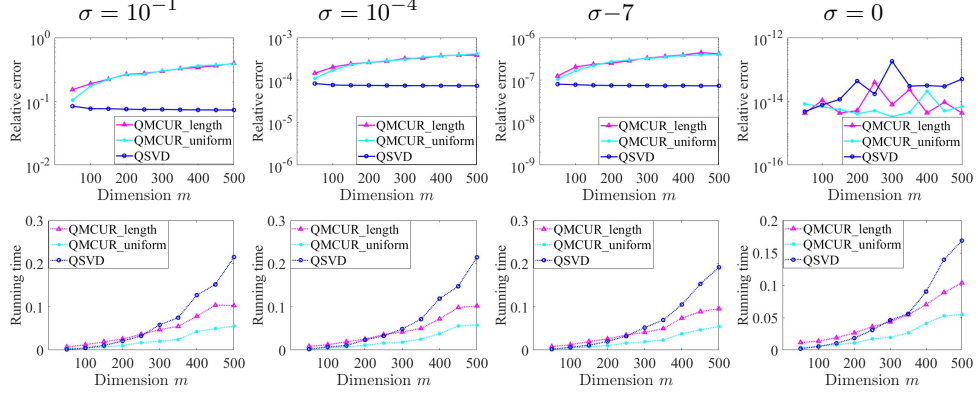


Fig. 2 Comparison of low rank quaternion matrix approximation methods under different noise levels σ . Rank 10 is used in all tests and m varies from 50 to 500. Top row: relative approximation errors vs. quaternion matrix dimensions. Bottom row: running time vs. quaternion matrix dimensions.

$$\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{E}. \quad (19)$$

In this experiment, we assume that $m = 500$ and $k = 50$. By applying Algorithm 1 to compute the rank- k approximation of $\tilde{\mathbf{X}}$ with $\sigma = 10^{-1}, 10^{-2}, \dots, 10^{-6}$, Fig. 1 shows the results. From this, we observe that the error $\|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{U}}\tilde{\mathbf{R}}\|_2$ increases linearly as $\|\mathbf{E}\|_2$ increases, which confirms the conclusion of Theorem 1.

Example 4.2 In this experiment, through simulation data, we will show the efficiency and accuracy of the QMCUR method by comparison with the QSVD method. We also consider the perturbed quaternion matrix $\tilde{\mathbf{X}}$ in Example 4.1. For all methods, the relative error $\|\mathbf{B} - \mathbf{X}\|_F / \|\mathbf{X}\|_F$ and the running time (in second) are used as metrics to measure approximate solutions, where $\mathbf{B}$ is obtained by QSVD, QMCUR_length, and QMCUR_uniform.

In Fig. 2, we present the relative error and running time for the synthetic data with varying size m , where the rank k is set to 10. One can see that both variations of the proposed QMCUR approximation are significantly faster than QSVD when m is sufficiently large. In the noiseless case (i.e., when $\sigma = 0$), the errors for QMCUR_uniform are smaller than those obtained by the QSVD algorithm. However, when additive noise appears, the proposed methods exhibit slightly worse but still good approximation accuracy. In summary, the proposed

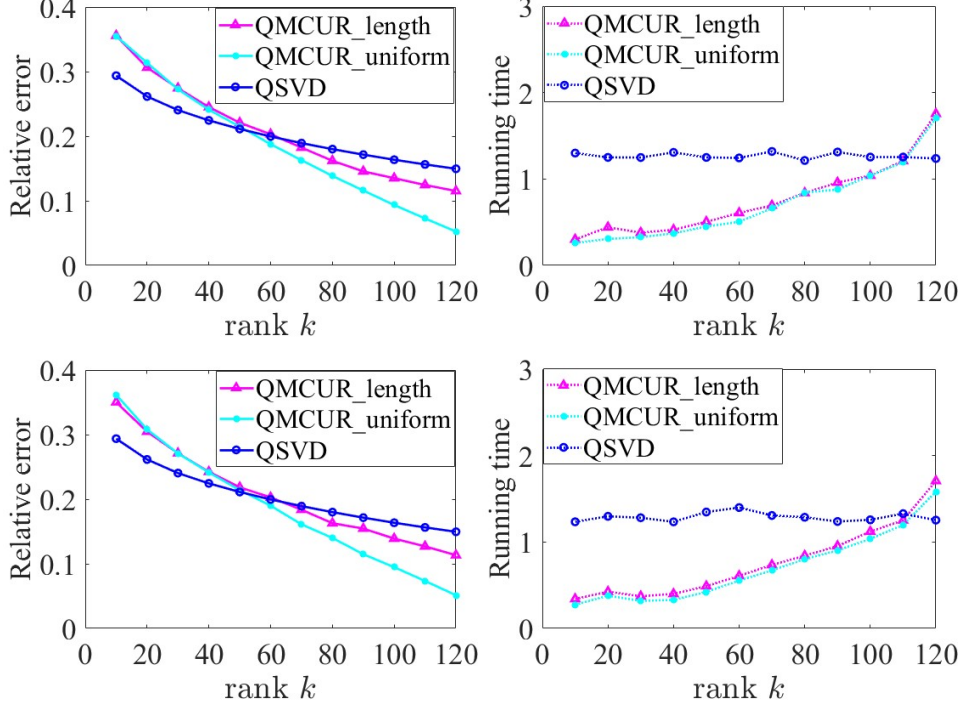


Fig. 3 Numerical simulation results of QSVD and QMCUR with different sampling strategies to color images with different rank k . From top to bottom: Image01 and Image02, respectively.

methods achieve a substantial speed-up while maintaining accuracy compared to QSVD, especially when m is sufficiently large.

Example 4.3 In this example, we evaluate the performance of the proposed QMCUR method for color image compression. The dataset used in this example is the Set27¹ and two images are shown in Fig. 4. Since the size of all images is $683 \times 1024 \times 3$, each image can be represented by a pure quaternion matrix $\mathbf{X} = \mathbf{R}\mathbf{i} + \mathbf{G}\mathbf{j} + \mathbf{B}\mathbf{k}$, where $\mathbf{R}$, $\mathbf{G}$, and $\mathbf{B}$ represent the RGB (red, green, blue) channels of the color image. The elements of $\mathbf{X}$ are $x_{ij} = r_{ij}\mathbf{i} + g_{ij}\mathbf{j} + b_{ij}\mathbf{k}$, where r_{ij} , g_{ij} , and b_{ij} denote the red, green, and blue pixel values at the position (i, j) of the color image, respectively.

Fig. 3 illustrates the relative error and running time of all methods for various selected ranks. From Fig. 3, we can observe that the performance of the proposed methods is superior to that of QSVD in terms of relative error when k is set to a suitable size. In terms of running time, the proposed methods are much faster than QSVD. In Fig. 4, we present the reconstructed image quality for various target ranks. It is evident that there is little difference in the visual effect between each original image and the approximate image obtained by the proposed methods and the QSVD.

Example 4.4 In this part, the proposed models are applied to the recovery of color images. The problem of recovering color images can be formulated as a quaternion matrix completion

¹<https://github.com/Huang-chao-yan/dataset>.

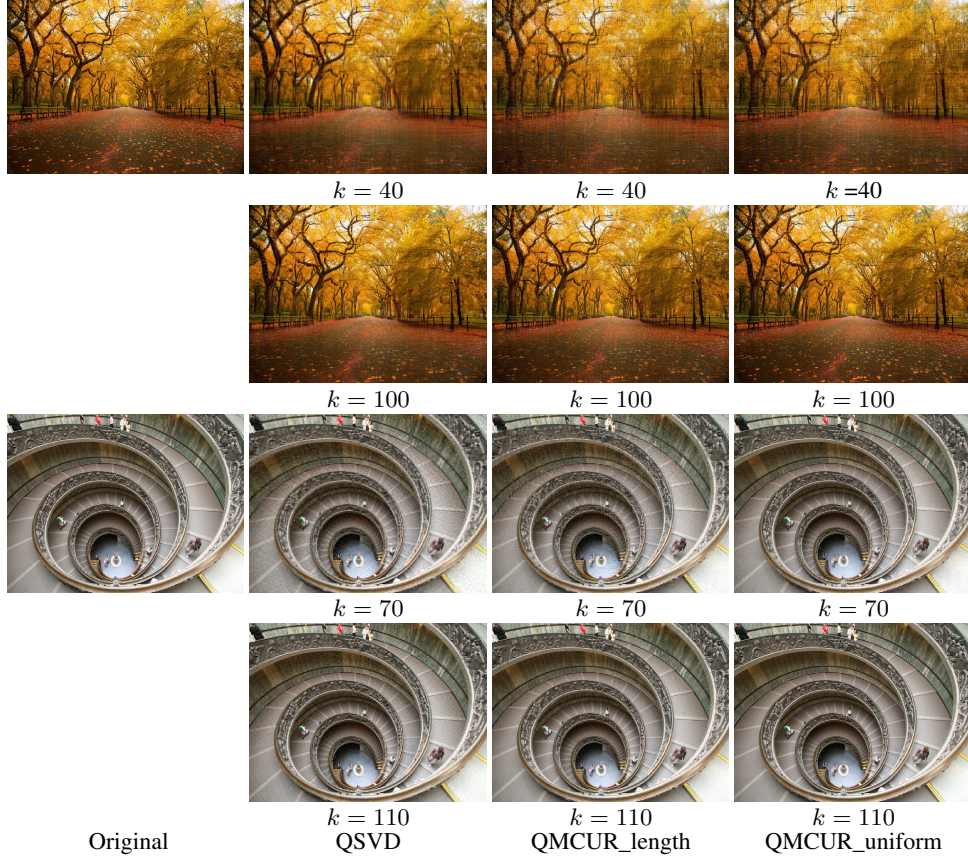


Fig. 4 Low-rank color image reconstruction. From top to bottom: Image01 and Image02, respectively.

problem, aiming to find its missing entries through the following optimization.

$$\min_{\mathbf{X}} \text{rank}(\mathbf{X}), \text{ s.t. } P_{\Omega}(\mathbf{X}) = P_{\Omega}(\mathbf{Y}), \quad (20)$$

where $\text{rank}(\cdot)$ denotes the rank function, $\mathbf{X} \in \mathbb{H}^{I_1 \times I_2}$ and $\mathbf{Y} \in \mathbb{H}^{I_1 \times I_2}$ represent the recovered and observed quaternion matrices, respectively. Ω represents the set of observed elements, and $P_{\Omega}(\mathbf{X})$ is a projection operator defined as $P_{\Omega}(\mathbf{X})_{ij} = \mathbf{X}_{ij}$ if $(i, j) \in \Omega$ and 0 otherwise. The rank minimization problem of a quaternion matrix can be expressed as the quaternion nuclear norm minimization (QNNM) [23]. However, the algorithm that utilizes QNNM involves the QSVD of quaternion matrices, which results in high time complexity. The QNNM method can be more effectively replaced by the concept of quaternion decomposition. The quaternion matrix decomposition variant of formulation (20) is presented as follows

$$\min_{\mathbf{X}} \|P_{\Omega}(\mathbf{X}) - P_{\Omega}(\mathbf{Y})\|_F^2, \text{ s.t. } \text{rank}(\mathbf{X}) = k, \quad (21)$$

where we assume that the unknown quaternion matrix $\mathbf{X}$ has a low-rank matrix representation. By introducing an auxiliary tensor variable $\mathbf{M}$, the optimization problem (21) can be

Algorithm 2 Quaternion matrix-CUR approximation from partially observed entries

Input: An incomplete quaternion matrix $\mathbf{Y} \in \mathbb{H}^{m \times n}$, observation locations Ω , rows and columns (i.e., I and J) indices that define $\mathbf{R} \in \mathbb{H}^{I \times n}$ and $\mathbf{C} \in \mathbb{H}^{m \times J}$ respectively, and maximal iteration $t_{max}=200$.

Output: The recovered quaternion matrix $\mathbf{X}^*$.

- 1: $\mathbf{X}^0 = \mathbf{Y} \in \mathbb{H}^{I \times J}$ is the observed quaternion matrix with missing pixels.
 - 2: $\mathbf{M}^0 \in \mathbb{H}^{I \times J}$ is a zero quaternion matrix.
 - 3: **For** $t = 0, 1, 2, \dots$ **do**
 - 4: $\mathbf{M}^{t+1} \leftarrow$ compute QMCUR approximation of the data matrix $\mathbf{X}^t$ using Algorithm 1
 - 5: $\mathbf{X}^{t+1} = (\mathbf{M}^{t+1})_{\Omega^c} + \mathbf{Y}_{\Omega}$, where Ω^c denotes the complementary set of Ω
 - 6: **If** $\|\dot{\mathbf{X}}^{t+1} - \dot{\mathbf{X}}^t\|_F / \|\dot{\mathbf{X}}^t\|_F \leq 10^{-4}$, or $t \geq t_{max}$ **then**
 - 7: $\mathbf{X}^* = \mathbf{X}^{t+1}$ and **break**
 - 8: **end**
 - 9: **end**
-

expressed as follows

$$\min_{\mathbf{X}} \|\mathbf{X} - \mathbf{M}\|_F^2, \quad \text{s.t. } \text{rank}(\mathbf{X}) = k, \quad P_{\Omega}(\mathbf{X}) = P_{\Omega}(\mathbf{Y}). \quad (22)$$

Therefore, we can alternatively solve the optimization problem (21) over the variables $\mathbf{X}$ and $\mathbf{M}$. The solution to the minimization problem (21) can be approximated using the following iterative procedures.

$$\mathbf{M}^t \leftarrow \mathcal{L}(\mathbf{X}^t), \quad (23)$$

$$\mathbf{X}^{t+1} \leftarrow \Omega \circledast \mathbf{Y} + (\mathbf{1} - \Omega) \circledast \mathbf{M}^t, \quad (24)$$

where t represents the iteration number, $\mathcal{L}$ is an operator used to calculate a low-rank quaternion matrix approximation of the quaternion matrix $\mathbf{X}$, which can be achieved by the QMCUR technique, and $\mathbf{1}$ is a matrix in which all components are equal to one. The algorithm consists of two main steps: low-rank quaternion matrix approximation (Eq. 23) and masking computation (Eq. 24). We summarize the proposed method in Algorithm 2. It starts with the initial incomplete data $\mathbf{X}^0$ and iteratively enhances the approximate solution until a stopping criterion is satisfied or the maximum number of iterations is reached.

We test our algorithm using four-color images of size $512 \times 768 \times 3$ from the Kodak PhotoCD Dataset² under the random missing scenario. The missing ratio (MR) is defined as the ratio of the number of missing elements to the total number of elements. The quality of the restoration results is measured using peak signal-to-noise rate (PSNR), structural similarity (SSIM), and running time per iteration (in seconds). Several low-rank matrix completion methods are compared in the experiments, including LRQA-1 [23], QLNF [26], and Q-DFN [29]. The numerical results of the color images recovered using various methods for different MRs are presented in Table 1, and the visualized results are depicted in Fig. 5.

From these results, the following observations are made. First, the PSNR results of the proposed methods (QMCUR_length and QMCUR_uniform) and three other methods (LRQA-1, QLNF, and Q-DFN) indicate that QMCUR_length and QMCUR_uniform achieve the best recovery performance in most cases. Furthermore, the proposed QMCUR_uniform is

²<http://r0k.us/graphics/kodak/>

Table 1 PSNR, SSIM, and running time per iteration (in second) of results by different methods with different MRs on color images. The best and second best values are respectively highlighted in boldface and underlined.

Data	Method	MR=90%			MR=80%			MR=70%		
		PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time
 Image03	Observed	9.18	0.038	0.000	9.69	0.104	0.000	10.26	0.184	0.000
	LRQA-1	24.71	0.954	0.460	26.73	<u>0.967</u>	0.446	28.07	<u>0.975</u>	0.503
	DFN	24.89	<u>0.953</u>	0.348	27.04	<u>0.967</u>	0.280	28.51	0.976	0.350
	QLNF	24.63	0.949	0.738	26.53	0.964	1.178	27.98	0.972	3.999
	QMCUR_length	25.12	0.951	<u>0.226</u>	<u>27.20</u>	<u>0.967</u>	<u>0.257</u>	<u>28.63</u>	0.976	<u>0.316</u>
	QMCUR_uniform	<u>24.99</u>	0.950	0.159	27.23	0.968	0.166	28.66	0.976	0.265
 Image04	Observed	8.00	0.024	0.000	8.50	0.050	0.000	9.09	0.082	0.000
	LRQA-1	22.79	<u>0.771</u>	0.456	25.50	0.868	0.498	27.24	0.905	0.478
	DFN	23.10	0.757	0.270	25.96	0.852	0.350	27.79	<u>0.889</u>	0.326
	QLNF	23.25	0.772	1.188	25.87	0.854	2.727	27.15	0.856	3.768
	QMCUR_length	<u>23.49</u>	0.776	<u>0.184</u>	<u>26.23</u>	<u>0.858</u>	<u>0.321</u>	<u>27.88</u>	<u>0.889</u>	<u>0.346</u>
	QMCUR_uniform	23.67	0.776	0.135	26.36	<u>0.848</u>	0.277	27.98	0.883	0.322
 Image05	Observed	7.22	0.022	0.000	7.73	0.045	0.000	8.30	0.072	0.000
	LRQA-1	21.85	0.725	0.478	23.98	0.798	0.451	25.41	0.841	0.542
	DFN	21.86	<u>0.702</u>	0.362	24.13	<u>0.783</u>	0.335	<u>25.60</u>	<u>0.825</u>	0.349
	QLNF	21.85	0.690	1.483	23.83	0.764	1.583	18.75	0.648	2.867
	QMCUR_length	<u>22.04</u>	0.690	<u>0.220</u>	<u>24.20</u>	0.779	<u>0.349</u>	25.50	0.824	<u>0.351</u>
	QMCUR_uniform	22.15	0.688	0.165	24.22	0.769	0.283	25.62	0.822	0.247
 Image06	Observed	7.20	0.026	0.000	7.71	0.058	0.000	8.29	0.095	0.000
	LRQA-1	21.75	0.815	0.434	24.85	<u>0.894</u>	0.512	26.84	<u>0.927</u>	0.501
	DFN	22.58	0.830	0.286	25.46	0.900	0.318	27.57	0.930	0.362
	QLNF	22.37	0.810	1.310	25.02	0.883	2.789	26.44	0.907	4.110
	QMCUR_length	<u>22.76</u>	0.819	<u>0.230</u>	<u>25.50</u>	0.892	<u>0.283</u>	<u>27.51</u>	0.923	<u>0.341</u>
	QMCUR_uniform	22.98	<u>0.829</u>	0.168	25.60	<u>0.894</u>	0.227	<u>27.55</u>	0.924	0.327

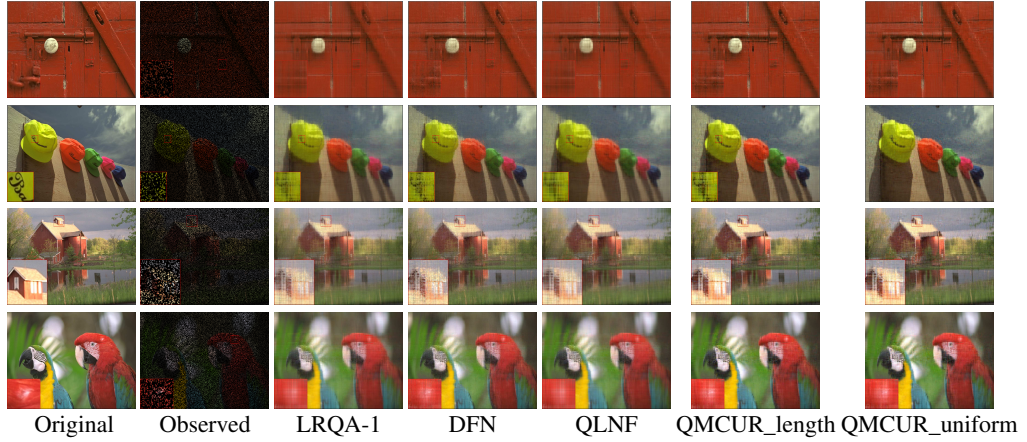


Fig. 5 Visual comparison of various methods for color image recovery. From top to bottom: Image03, Image04, Image05, and Image06, respectively.

the fastest, demonstrating its scalability on large-scale color images. As shown in Fig. 5, the visual effect of the QMCUR_length and QMCUR_uniform methods is superior to the comparative methods with $MR = 80\%$, which is consistent with our quantitative results. Both of these findings further demonstrate the superiority of our methods over comparative methods for handling large-scale data.

5 Conclusion

This work presented an efficient quaternion matrix CUR method for computing low-rank approximation of a quaternion data matrix. The QMCUR approximation offers a balance between accuracy and computational costs by selecting specific column and row submatrices of a given quaternion matrix. Next, we conducted a perturbation analysis of this approximation and concluded that the error in the quaternion spectral norm is correlated with the noise quaternion matrix in the first order. Computational experiments illustrated that the QMCUR approximation methods are significantly faster than comparative low-rank quaternion matrix approximation methods, without sacrificing the quality of reconstruction on both synthetic and color image datasets.

Acknowledgments. This research is supported by University of Macau (MYRG2022-00108-FST, MYRG-CRG2022-00010-ICMS), the Science and Technology Development Fund, Macau S.A.R (0036/2021/AGJ), the National Key Research Project (2022YFB3904104), and the Key Project from National Natural Science Foundation of China (42230406).

Declarations

Ethical Approval. Not applicable.

Availability of supporting data. The data sets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Competing interests. The authors declare no competing interests.

Funding. This research is supported by University of Macau (MYRG2022-00108-FST, MYRG-CRG2022-00010-ICMS), the Science and Technology Development Fund, Macau S.A.R (0036/2021/AGJ), the National Key Research Project (2022YFB3904104), and the Key Project from National Natural Science Foundation of China (42230406).

Authors contributions. Pengling Wu: Methodology, Software, Writing - Original Draft. Kit Ian Kou: Conceptualization, Validation, Resources. Hongmin Cai: Methodology, Validation, Writing - Review and Editing. Zhaoyuan Yu: Formal analysis, Writing - Review and Editing.

References

- [1] Hamilton, W.R.: Elements of Quaternions. Longmans, Green, & Company, (1866)

- [2] Cheng, D., Kou, K.I.: Generalized sampling expansions associated with quaternion fourier transform. *Mathematical Methods in the Applied Sciences* **41**(11), 4021–4032 (2018)
- [3] Ell, T.A., Le Bihan, N., Sangwine, S.J.: *Quaternion Fourier Transforms for Signal and Image Processing*. John Wiley & Sons, (2014)
- [4] Le Bihan, N., Mars, J.: Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing. *Signal Processing* **84**(7), 1177–1199 (2004)
- [5] Hirose, A., Yoshida, S.: Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. *IEEE Transactions on Neural Networks and Learning Systems* **23**(4), 541–551 (2012)
- [6] Zhou, H., Zhang, X., Zhang, C., Ma, Q.: Quaternion convolutional neural networks for hyperspectral image classification. *Engineering Applications of Artificial Intelligence* **123**, 106234 (2023)
- [7] Zhang, M., Ding, W., Li, Y., Sun, J., Liu, Z.: Color image watermarking based on a fast structure-preserving algorithm of quaternion singular value decomposition. *Signal Processing* **208**, 108971 (2023)
- [8] Li, M., Yuan, X., Chen, H., Li, J.: Quaternion discrete fourier transform-based color image watermarking method using quaternion qr decomposition. *IEEE Access* **8**, 72308–72315 (2020)
- [9] Liu, W., Kou, K.I., Miao, J., Cai, Z.: Quaternion scalar and vector norm decomposition: Quaternion pca for color face recognition. *IEEE Transactions on Image Processing* **32**, 446–457 (2023)
- [10] Wang, G., Zhang, D., Vasiliev, V.I., Jiang, T.: A complex structure-preserving algorithm for the full rank decomposition of quaternion matrices and its applications. *Numerical Algorithms* **91**(4), 1461–1481 (2022)
- [11] Dong Zhang, C.J. Tongsong Jiang, Wang, G.: A complex structure-preserving algorithm for computing the singular value decomposition of a quaternion matrix and its applications. *Numerical Algorithms* **95**(4), 267–283 (2024)
- [12] Qi, L., Luo, Z., Wang, Q.-W., Zhang, X.: Quaternion matrix optimization: Motivation and analysis. *Journal of Optimization Theory and Applications* **193**(1-3), 621–648 (2022)
- [13] Zou, C., Kou, K.I., Wang, Y.: Quaternion collaborative and sparse representation with application to color face recognition. *IEEE Transactions on Image Processing* **25**(7), 3287–3302 (2016)
- [14] Zou, C., Kou, K.I., Dong, L., Zheng, X., Tang, Y.Y.: From grayscale to color: Quaternion linear regression for color face recognition. *IEEE Access* **7**, 154131–154140 (2019)
- [15] Chen, J., Ng, M.K.: Color image inpainting via robust pure quaternion matrix completion: Error bound and weighted loss. *SIAM Journal on Imaging Sciences* **15**(3), 1469–1498 (2022)
- [16] Jia, Z., Ng, M.K., Song, G.: Robust quaternion matrix completion with applications to image inpainting. *Numerical Linear Algebra with Applications* **26**(4), 2245 (2019)
- [17] Li, C., Liu, Y., Wu, F., Che, M.: Randomized block krylov subspace algorithms for low-rank quaternion matrix approximations. *Numerical Algorithms*, 1–31 (2023)
- [18] Gai, S., Yang, G., Wan, M., Wang, L.: Denoising color images by reduced quaternion matrix singular value decomposition. *Multidimensional Systems and Signal Processing*

- 26**, 307–320 (2015)
- [19] Huang, C., Li, Z., Liu, Y., Wu, T., Zeng, T.: Quaternion-based weighted nuclear norm minimization for color image restoration. *Pattern Recognition* **128**, 108665 (2022)
 - [20] Jia, Z., Jin, Q., Ng, M.K., Zhao, X.: Non-local robust quaternion matrix completion for large-scale color image and video inpainting. *IEEE Transactions on Image Processing* **31**, 3868–3883 (2022)
 - [21] Chen, J., Ng, M.K.: Color image inpainting via robust pure quaternion matrix completion: Error bound and weighted loss. *SIAM Journal on Imaging Sciences* **15**(3), 1469–1498 (2022)
 - [22] Xu, T., Kong, X., Shen, Q., Chen, Y., Zhou, Y.: Deep and low-rank quaternion priors for color image processing. *IEEE Transactions on Circuits and Systems for Video Technology* **33**(7), 3119–3132 (2023)
 - [23] Chen, Y., Xiao, X., Zhou, Y.: Low-rank quaternion approximation for color image processing. *IEEE Transactions on Image Processing* **29**, 1426–1439 (2020)
 - [24] Yu, Y., Zhang, Y., Yuan, S.: Quaternion-based weighted nuclear norm minimization for color image denoising. *Neurocomputing* **332**, 283–297 (2019)
 - [25] Yang, L., Kou, K.I., Miao, J.: Weighted truncated nuclear norm regularization for low-rank quaternion matrix completion. *Journal of Visual Communication and Image Representation* **81**, 103335 (2021)
 - [26] Yang, L., Miao, J., Kou, K.I.: Quaternion-based color image completion via logarithmic approximation. *Information Sciences* **588**, 82–105 (2022)
 - [27] Miao, J., Kou, K.I.: Color image recovery using low-rank quaternion matrix completion algorithm. *IEEE Transactions on Image Processing* **31**, 190–201 (2022)
 - [28] Ren, H., Ma, R.-R., Liu, Q., Bai, Z.-J.: Randomized quaternion qlp decomposition for low-rank approximation. *Journal of Scientific Computing* **92**(3), 80 (2022)
 - [29] Miao, J., Kou, K.I.: Quaternion-based bilinear factor matrix norm minimization for color image inpainting. *IEEE Transactions on Signal Processing* **68**, 5617–5631 (2020)
 - [30] Liu, Q., Ling, S., Jia, Z.: Randomized quaternion singular value decomposition for low-rank matrix approximation. *SIAM Journal on Scientific Computing* **44**(2), 870–900 (2022)
 - [31] Goreinov, S.A., Tyrtyshnikov, E.E., Zamarashkin, N.L.: A theory of pseudoskeleton approximations. *Linear Algebra and its Applications* **261**(1), 1–21 (1997)
 - [32] Aldroubi, A., Hamm, K., Koku, A.B., Sekmen, A.: Cur decompositions, similarity matrices, and subspace clustering. *Frontiers in Applied Mathematics and Statistics* **4** (2019)
 - [33] Lin, P., Peng, S., Xiang, Y., Li, C., Cui, X., Zhang, W.: Structure-oriented cur low-rank approximation for random noise attenuation of seismic data. *IEEE Transactions on Geoscience and Remote Sensing* **61**, 1–13 (2023)
 - [34] Drineas, P., Mahoney, M.W., Muthukrishnan, S.: Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications* **30**(2), 844–881 (2008)
 - [35] Wang, S., Zhang, Z.: Improving cur matrix decomposition and the nystrom approximation via adaptive sampling. *The Journal of Machine Learning Research* **14**(1), 2729–2769 (2013)
 - [36] Zhang, F.: Quaternions and matrices of quaternions. *Linear Algebra and its Applications* **251**, 21–57 (1997)

- [37] Ling, C., He, H., Qi, L.: Singular values of dual quaternion matrices and their low-rank approximations. *Numerical Functional Analysis and Optimization* **43**(12), 1423–1458 (2022)
- [38] Wang, Q.-W.: The general solution to a system of real quaternion matrix equations. *Computers & Mathematics with Applications* **49**(5), 665–675 (2005)
- [39] Drineas, P., Kannan, R., Mahoney, M.W.: Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing* **36**(1), 184–206 (2006)
- [40] Chiu, J., Demanet, L.: Sublinear randomized algorithms for skeleton decompositions. *SIAM Journal on Matrix Analysis and Applications* **34**(3), 1361–1383 (2013)