# Low-Rank Quaternion Approximation for Color Image Processing

Yongyong Chen, Xiaolin Xiao, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*— Low-rank matrix approximation (LRMA)-based methods have made a great success for grayscale image processing. When handling color images, LRMA either restores each color channel independently using the monochromatic model or processes the concatenation of three color channels using the concatenation model. However, these two schemes may not make full use of the high correlation among RGB channels. To address this issue, we propose a novel low-rank quaternion approximation (LRQA) model. It contains two major components: first, instead of modeling a color image pixel as a scalar in conventional sparse representation and LRMA-based methods, the color image is encoded as a pure quaternion matrix, such that the cross-channel correlation of color channels can be well exploited; second, LRQA imposes the low-rank constraint on the constructed quaternion matrix. To better estimate the singular values of the underlying low-rank quaternion matrix from its noisy observation, a general model for LRQA is proposed based on several nonconvex functions. Extensive evaluations for color image denoising and inpainting tasks verify that LRQA achieves better performance over several state-of-the-art sparse representation and LRMA-based methods in terms of both quantitative metrics and visual quality.**

***Index Terms*— Low-rank matrix approximation, image denoising, image inpainting, quaternion singular value decomposition, nonconvex approximation.**

## I. Introduction

As an emerging mathematical tool, low-rank matrix approximation (LRMA) has been applied in a broad family of real applications, such as image denoising [1]–[3], image inpainting [4], [5], image deblurring [6], background-foreground separation [1], [7], subspace clustering [8], and so on. Due to the fact that the high-dimensional data inherently possess a low-rank structure [9], the goal of LRMA is to exactly and efficiently recover the underlying low-rank matrix from its corrupted observation [1].

Manuscript received March 28, 2019; revised August 8, 2019; accepted September 5, 2019. Date of publication September 19, 2019; date of current version November 7, 2019. This work was funded in part by the Science and Technology Development Fund, Macau SAR (File no. 189/2017/A3), and by the Research Committee at University of Macau under Grants MYRG2016-00123-FST and MYRG2018-00136-FST. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nikolaos Mitianoudis. *(Corresponding author: Yicong Zhou.)*

Y. Chen and Y. Zhou are with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yongyongchen.cn@gmail.com; yicongzhou@um.edu.mo).

X. Xiao is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: shellyxiaolin@gmail.com).

Digital Object Identifier 10.1109/TIP.2019.2941319

TABLE I

Summary of Existing Nonconvex Surrogate Functions $\phi(x, \gamma)$

| Function name | $\phi(x,\gamma)$ | Function name | $\phi(x,\gamma)$ |
|---|---|---|---|
| Schatten-$\gamma$ [18] | $x^\gamma$ | Weighted Schatten-$\gamma$ [2] | $\omega x^\gamma$ |
| Logarithm [45] | $\log(\gamma + x)$ | Log-determinant [19] | $\log(1 + x^2)$ |
| Laplace [3] | $1 - e^{\frac{-x}{\gamma}}$ | Geman [17] | $\frac{(1+\gamma)x}{\gamma+x}$ |

Most of LRMA-based variants follow two lines [2], [10], [11]: matrix factorization and matrix rank minimization. In the first line, it usually factorizes the matrix to be recovered into two or three thin matrices. An intuitive example is the popular nonnegative matrix factorization [12]. It factorizes the original data matrix into two factor matrices with nonnegative constraints. And their product can well approximate the original data matrix under the Frobenius-norm fidelity. Recently, the work in [13] proposed a bilinear factor matrix norm minimization for LRMA. The main issue of the matrix factorization-based LRMA algorithms is lack of the rank values in real applications [3], [10], [14]. The second line is usually achieved by different rank approximation regularizers, such as the nuclear norm [15]. Candès and Recht [16] has proven that the nuclear norm is the tightest convex relaxation of the NP-hard rank minimization function. However, many works [1]–[3], [17] have pointed out that the nuclear norm-based LRMA may yield sub-optimal results of the original rank minimization. The reason is that each singular value is treated equally, which is contrary to the fact that the large singular values may contain more information of the original data. To better approximate the rank function, many nonconvex surrogates have been proposed, such as the Schatten $\gamma$-norm [18], weighted nuclear norm [1], log-determinant penalty [19] (summarized in Table I) and achieved the promising performance in grayscale image processing. Among them, the most popular one is the weighted nuclear norm (WNNM), which assigns different weights to different singular values by considering the physical significance of them. Following this line, Xie *et al*. [2] proposed the weighted Schatten $\gamma$-norm minimization (WSNM) by combining WNNM and the Schatten $\gamma$-norm for image denoising and background-foreground separation. In [20], the authors proposed a generalized nonconvex nonsmooth LRMA solver, named iteratively reweighted nuclear norm algorithm. To promote the applicability of these nonconvex surrogates, Nie *et al*. [21]

proposed a parameter-free nonconvex LRMA model for image inpainting.

When handling color images using LRMA, we may encounter two challenges. The first challenge is ***how to fully exploit the high correlation among RGB channels***. An intuitive way is to perform LRMA channel-by-channel and then combine all results together. That is, LRMA is applied to each channel independently. This scheme is termed as the monochromatic model. An obvious shortcoming of the monochromatic model is that the correlation among three channels are completely ignored, resulting in performance degradation in color image processing. Another popular way is to concatenate three RGB channels to make use of the channel correlation. This scheme is named as the concatenation model [22]. For example, Xu *et al*. [23] proposed a multi-channel weighted nuclear norm minimization model. It first vectorizes each similar patch and then concatenates all vectors to form a noisy patch matrix. From the perspective of tensor-unfolding [9], the concatenation model-based color image processing methods just use one unfolding matrix and ignore other two unfolding matrices. In summary, these above two schemes may fail to make full use of the correlation information among three color channels.

The second challenge is ***how to preserve the local details of color images under the LRMA framework***. In addition to the channel correlation, the nonlocal self-similarity (NSS) [24] prior is another intrinsic characteristic for image processing tasks, such as image denoising, image deblurring. The underlying assumption of NSS is that for an exemplar patch, there are many similar local patches across a natural image [1], [24]. For example, Zhang *et al*. [25] used principal component analysis on the similar patches group to remove the Gaussian noise. Combining the sparse representation and the NSS prior, the work in [26] proposed a nonlocally centralized sparse representation model for image restoration. Borrowing the above idea, the advanced version of sparse representation, LRMA as well as NSS was fully considered in the works of [1], [2].

The above-mentioned methods may cause unsatisfactory results for color image processing since they were originally designed for grayscale image processing. Recently, the quaternion representation (QR)-based color image processing methods have attracted much attention due to its rapid development in both theory [27]–[29] and applications [22], [30]–[36]. Due to the essential characteristic of QR which includes one real part and three imaginary components, it perfectly fits color image processing tasks. Using QR, a color image is represented by a pure quaternion matrix instead of three independent matrices or a concatenation matrix. This means that all color channels are handled simultaneously in the quaternion domain, resulting in the superior power of preserving cross-channel correlations of color images. For example, inspired by the sparse representation [37], Xu *et al*. [22] developed a vector sparse representation model for color image denoising, inpainting, and super-resolution. The work in [33] extended the sparse representation and collaborative representation classifications into the quaternion domain and achieved good performance for color face recognition. In [35], Zhu *et al*. proposed a quaternion convolution neural network (QCNN) model to extend CNN from the real domain into the quaternion domain. It contains a series of basic models like quaternion convolution layer. To explore the fundamental reason of the success of QCNN, the work in [38] used the Hamilton product to capture internal latent relations within features. Following [35], Zeng *et al*. [34] proposed the quaternion principal component analysis network for color image classification. To preserve the spatial structures of images, Xiao and Zhou [30] proposed a two-dimensional quaternion sparse principal component analysis method for color image recognition. Generally speaking, the main advantage of QR over the real valued methods is the generalization capability [22], [30], [38].

Considering the huge success of LRMA and the powerful data representation advance of QR, one natural question is that ***is it possible and reasonable to extend LRMA into the quaternion domain to handle the three color channels in a holistic manner?*** Therefore, in this paper, we propose a novel model, *i.e.*, low-rank quaternion approximation (LRQA) with applications to color image denoising and color image inpainting. Our idea is motivated by the recently proposed LRMA [2], [3], [17] and the quaternion representation-based methods in [22], [30], and [33]. However, different from these methods in [2], [3], [17], LRQA incorporates the quaternion representation into LRMA to handle three color channels as a whole for the better capture of the high correlation. A general model for LRQA based on three nonconvex functions is proposed. To derive the closed-form solution of LRQA, we first compute the complex singular value decomposition (SVD) instead of quaternion SVD to reduce the high computation cost and then adopt the difference of convex method to linearize three nonconvex functions. The main contributions of this paper are summarized as follows:

- We propose a novel model, namely, low-rank quaternion approximation (LRQA) for color image processing. Different from LRMA-based methods treating color image pixel as a scalar, LRQA encodes each color image pixel as a pure quaternion number, such that more correlation information among RGB channels can be well preserved.
- To achieve a better quaternion rank estimation, we propose a general model for LRQA based on three nonconvex functions and derive a general solver.
- LRQA is further extended to handle the color image denoising and inpainting tasks. The experimental results on both simulated and real color images demonstrate their superiority over several state-of-the-art methods.

The rest of this paper is structured as follows. Section II briefly reviews related works on LRMA and quaternion-based methods. In Section III, we discuss the basics of quaternion representation. Section IV proposes LRQA in detail and analyzes the general solver for LRQA. Two real applications of LRQA are described in Section V. We evaluate the performance of the proposed LRQA in Section VI and conclude this paper in Section VII.

## II. Related Work

In this section, we briefly review the highly related topics: low-rank matrix approximation and quaternion-based methods.

### A. Low-Rank Matrix Approximation

Given a real matrix $Y \in \mathbb{R}^{m\times n}$, LRMA can be formulated as the following minimization problem:

$$\min_X \frac{1}{2}\|X - Y\|_F^2 + \lambda * rank(X), \tag{1}$$

where $\lambda$ is a nonnegative parameter and $rank(\cdot)$ is the rank function. In Eq. (1), the first term is the data fidelity term which aims to measure the difference between $X$ and $Y$, while the second one is called regularizer, such as the nuclear norm $\|X\|_* = \sum_i \sigma_i$. A large number of extensions of LRMA have been presented, including robust principal component analysis [39], matrix completion [40], low-rank representation [8], [41], robust matrix sensing [42] and so on. To address the essential issue of the nuclear norm, truncated nuclear norm [4], capped nuclear norm [43], partial sum nuclear norm [44], weighted nuclear norm [1], and weighted Schatten $\gamma$-norm [2] have successively been proposed as summarized in Table I. In a word, the main idea of these nonconvex approximations is that no matter how big a singular value is, its contribution for the rank function should be 1.

The existing LRMA-based approaches work well for grayscale images, but degrade sharply when handling color images [23]. The main reasons are that, one the one hand, these approaches handle each color channel in a channel-by-channel manner, which means that each color channel is treated as an independent grayscale image. On the other hand, they process the concatenation of three color channels. As discussed in the introduction part, however, it is still intractable to preserve the correlation information among RGB channels.

### B. Quaternion-Based Methods

Quaternion-based methods have received increasing attention in various applications, such as face recognition [33], [46], dimension reduction [30], image denoising [22], since it can treat a color image as a whole, such that the cross-channel correlation of color images can be well encoded. Considering the fact that an image signal can be sparsely represented by a linear combination of the dictionary atoms, Xu *et al*. [22] developed a quaternion-based sparse representation model for several color image processing tasks. Following this line, Zou *et al*. [33] proposed two quaternion methods named quaternion collaborative and sparse representation-based classification models for color face recognition, which aims to learn the collaborative or sparse coefficients of a testing sample, and defined specific operators to convert the quaternion variables into real domain.

Essentially, the proposed LRQA is different from [22], [33] since the methods in [22] and [33] belong to the one-dimensional vector sparse representation methods, while our proposed LRQA belongs to the two-dimensional low-rank matrix approximation method, which imposes the sparsity on the singular values of the quaternion matrix.

## III. Preliminaries

In this section, we briefly review some mathematical notations and definitions for quaternion number and quaternion representation. In this paper, scalars, vectors, matrices, and tensors are denoted as lowercase letters, boldface lowercase letters, boldface capital letters, and boldface Euler script letters, *e.g.*, $x$, $\mathbf{x}$, $\mathbf{X}$, and $\mathcal{X}$, respectively. Following [22], [31], a dot above the variable (*e.g.*, $\dot{\mathbf{x}} \in \mathbb{H}$) is to denote the variable in quaternion space $\mathbb{H}$.

As a generalization of the real space $\mathbb{R}$ and complex space $\mathbb{C}$, the quaternion space was firstly introduced by Hamilton [47]. A quaternion number $\dot{x}$ includes one real number and three imaginary parts, *i.e.*,

$$\dot{x} = x_0 + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k}, \tag{2}$$

where $x_l \in \mathbb{R}$ $(l = 0, 1, 2, 3)$ is real number, and $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are three imaginary units. If $x_0 = 0$, $\dot{x}$ is called the pure quaternion.

Several important definitions of norm, properties and operator of quaternion numbers and matrices are introduced as following: Given a quaternion number $\dot{x}$ and a quaternion matrix $\dot{\mathbf{X}} \in \mathbb{H}^{m\times n}$,

- The conjugate operator $\overline{\dot{x}}$, the modulus $|\dot{x}|$, and the inverse $\dot{x}^{-1}$ of $\dot{x}$: $\overline{\dot{x}} = x_0 - (x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k})$, $|\dot{x}| = \sqrt{x_0^2 + x_1^2 + x_2^2 + x_3^2}$, $\dot{x}^{-1} = \frac{\dot{x}^*}{|\dot{x}|^2}$; The conjugate operator $\overline{\dot{\mathbf{X}}}$, the transpose operator $\dot{\mathbf{X}}^T$, the conjugate transpose $\dot{\mathbf{X}}^\triangleleft$: $\overline{\dot{\mathbf{X}}} = (\overline{\dot{x}_{i,j}})$, $\dot{\mathbf{X}}^T = (\dot{x}_{j,i})$, $\dot{\mathbf{X}}^\triangleleft = (\overline{\dot{\mathbf{X}}})^T = \overline{\dot{\mathbf{X}}^T} = (\overline{\dot{x}_{j,i}})$;
- The F-norm of $\dot{\mathbf{X}}$: $\|\dot{\mathbf{X}}\|_F = \big(\sum_{i=1}^{m}\sum_{j=1}^{n}|\dot{x}_{i,j}|^2\big)^{\frac{1}{2}}$.
- $\dot{\mathbf{X}} \in \mathbb{H}^{m\times m}$ is called unitary quaternion matrix if and only if $\dot{\mathbf{X}}\dot{\mathbf{X}}^\triangleleft = \dot{\mathbf{X}}^\triangleleft\dot{\mathbf{X}} = \mathbf{I}_m$ where $\mathbf{I}_m \in \mathbb{R}^{m\times m}$ is a real identity matrix.

*Definition 1 (Cayley-Dickson construction [48] and Complex adjoint form [28]):* For any quaternion matrix $\dot{\mathbf{X}} = \mathbf{X}_0 + \mathbf{X}_1\mathbf{i} + \mathbf{X}_2\mathbf{j} + \mathbf{X}_3\mathbf{k} \in \mathbb{H}^{m\times n}$, it can be represented by an ordered pair of complex matrices $\mathbf{X}_a$, $\mathbf{X}_b \in \mathbb{C}^{m\times n}$, *i.e.*,

$$\dot{\mathbf{X}} = \mathbf{X}_a + \mathbf{X}_b\mathbf{j}, \tag{3}$$

where $\mathbf{X}_a = \mathbf{X}_0 + \mathbf{X}_1\mathbf{i}$, $\mathbf{X}_b = \mathbf{X}_2 + \mathbf{X}_3\mathbf{i}$. Then the complex adjoint form of $\dot{\mathbf{X}}$ is formulated as

$$\chi_{\dot{\mathbf{X}}} = \begin{bmatrix} \mathbf{X}_a & \mathbf{X}_b \\ -\overline{\mathbf{X}}_b & \overline{\mathbf{X}}_a \end{bmatrix} \in \mathbb{C}^{2m\times 2n}. \tag{4}$$

Please refer to [28], [49] for detailed introduction of quaternion algebra.

In Section IV-B, a third-order tensor $\mathcal{X}$ of size $m \times n \times 3$ is transformed into a quaternion matrix $\dot{\mathbf{X}} \in \mathbb{H}^{m\times n}$ by setting three frontal slices of $\mathcal{X}$ as the three imaginary parts of $\dot{\mathbf{X}}$, *i.e.*, $\dot{\mathbf{X}} = \mathcal{X}(:,:,1)\mathbf{i} + \mathcal{X}(:,:,2)\mathbf{j} + \mathcal{X}(:,:,3)\mathbf{k}$. In Section V, the estimated quaternion matrix $\dot{\mathbf{X}}$ is transformed back to a color image by extracting the three imaginary parts of $\dot{\mathbf{X}}$ as three color channels.

## IV. Low-Rank Quaternion Approximation

Considering that the existing LRMA-based methods may suffer from performance degradation of color image process-

ing since they are initially designed in the real settings for grayscale image processing, we propose the low-rank quaternion approximation (LRQA) for the better capture of the cross-channel relationship. We first solve two challenges of LRQA, and then propose the general framework of LRQA for better quaternion rank approximation.

### A. LRQA

Given a quaternion matrix $\dot{Y}$, the proposed LRQA is defined as

$$\min_{\dot{\mathbf{X}}} \frac{1}{2}\|\dot{\mathbf{X}} - \dot{\mathbf{Y}}\|_F^2 + \lambda * rank(\dot{\mathbf{X}}), \tag{5}$$

where $rank(\dot{\mathbf{X}})$ is the rank function of quaternion matrix $\dot{\mathbf{X}}$. As for how to construct the quaternion matrix $\dot{\mathbf{Y}}$, please refer to Section V.

Different from LRMA in real domain, LRQA in Eq. (5) poses at least two challenges for color image analysis: the first one is how to define the rank of a quaternion matrix; the other one is how to obtain the closed-form solution of quaternion rank regularized minimization problem. In the following section, we will give the affirmative answer to these above two questions. Our solution is inspired by the quaternion singular value decomposition [28], [29], [32].

*Theorem 1 ( [28] Quaternion singular value decomposition (QSVD)):* For any quaternion matrix $\dot{\mathbf{Y}} \in \mathbb{H}^{m\times n}$ of rank $r$, there exist a pair of unitary quaternion matrices $\dot{\mathbf{U}} \in \mathbb{H}^{m\times m}$ and $\dot{\mathbf{V}} \in \mathbb{H}^{n\times n}$ such that

$$\dot{\mathbf{Y}} = \dot{\mathbf{U}} \begin{pmatrix} \Sigma_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \dot{\mathbf{V}}^{\triangleleft}, \tag{6}$$

where $\triangleleft$ is the conjugate transpose operator and $\Sigma_r \in \mathbb{R}^{r\times r}$ is a real diagonal matrix.

The above QSVD first introduced by Zhang [28] in 1996 also enjoys similar forms and properties of the SVD for a real matrix. For example, all singular values of any quaternion matrix are nonnegative and have the decreasing order property. The bigger singular values also preserve the major color image information, such as the color information and the structural details. This observation can be found in [32]. Based on QSVD, we can give the following definition.

*Definition 2 (Quaternion rank):* The rank of a quaternion matrix can be defined as the number of nonzero singular values.

Until now, we have given the positive answer to the first challenge of Eq. (5). Actually, the quaternion rank is the $l_0$-norm of singular value vector $\{\sigma_i\}_{i=1}^{\min(m,n)}$. Unfortunately, the $l_0$-norm is nonconvex. Borrowing the idea of the nuclear norm in real domain, we consider the following quaternion nuclear norm:

*Definition 3 (Quaternion nuclear norm):* The quaternion nuclear norm of a quaternion matrix $\dot{\mathbf{X}}$ is defined as the sum of all nonzero singular values, *i.e.*, $\|\dot{\mathbf{X}}\|_{\circledast} = \Sigma_i \sigma_i(\dot{\mathbf{X}})$.

Intuitively, the nuclear norm is extended into the quaternion domain by *Definition* 3 and enjoys the same form since all singular values are still real numbers. By using the quaternion nuclear norm, the nonconvex optimization problem (5) can be relaxed as:

$$\min_{\dot{\mathbf{X}}} \frac{1}{2}\|\dot{\mathbf{Y}} - \dot{\mathbf{X}}\|_F^2 + \lambda\|\dot{\mathbf{X}}\|_{\circledast}. \tag{7}$$

Here, we call (7) as quaternion nuclear norm minimization (QNNM). The remaining problem is how to obtain the optimal solution of QNNM (7).

*Theorem 2: (Quaternion singular value thresholding (QSVT)*): For any $\lambda > 0$ and quaternion matrix $\dot{\mathbf{Y}}$, the closed-form solution of QNNM (7) is:

$$\dot{\mathbf{X}}^* = \dot{\mathbf{U}}\mathcal{S}_\lambda(\Sigma)\dot{\mathbf{V}}^{\triangleleft}, \tag{8}$$

where $\dot{\mathbf{Y}} = \dot{\mathbf{U}}\Sigma\dot{\mathbf{V}}^{\triangleleft}$ is the QSVD of $\dot{\mathbf{Y}}$. The soft thresholding operator is defined as $\mathcal{S}_\lambda(\Sigma) = diag(\max\{\sigma_i(\dot{\mathbf{Y}}) - \lambda, 0\})$.

The proof of *Theorem* 2 can be found in the supplementary material Appendix A. Essentially, our quaternion nuclear norm is the $l_1$-norm of the singular value vector, while the quaternion rank is the $l_0$-norm of the singular value vector. From the above theorem, we can see that QSVT shrinks each singular value equally, that is, each singular value is minus the same threshold. Therefore, the quaternion nuclear norm still over-penalizes larger singular values as well as the nuclear norm in real domain since $l_1$-norm is a biased estimator of the $l_0$-norm [1], [3]. This observation encourages us to seek a more accurate approximation of quaternion rank.

### B. General Model of LRQA

Motivated by the promising performance of nonconvex surrogates in LRMA, we propose a general model of LRQA as follows:

$$\min_{\dot{\mathbf{X}}} \frac{1}{2}\|\dot{\mathbf{Y}} - \dot{\mathbf{X}}\|_F^2 + \lambda \sum_i \phi\big(\sigma_i(\dot{\mathbf{X}}), \gamma\big), \tag{9}$$

where $\phi(\cdot)$ is defined in Table I. $\gamma$ is a nonnegative parameter related to the specific $\phi(\cdot)$. In particular, we consider three special cases:

$$\|\dot{\mathbf{X}}\|_{l,\gamma} = \sum_i (1 - e^{\frac{-\sigma_i(\dot{\mathbf{X}})}{\gamma}}); \tag{10}$$

$$\|\dot{\mathbf{X}}\|_{g,\gamma} = \sum_i \frac{(1+\gamma) * \sigma_i(\dot{\mathbf{X}})}{\gamma + \sigma_i(\dot{\mathbf{X}})}; \tag{11}$$

$$\|\dot{\mathbf{X}}\|_{\omega,\gamma}^{\gamma} = \sum_i \omega_i * \sigma_i^{\gamma}(\dot{\mathbf{X}}); \tag{12}$$

where $\omega_i$ is a nonnegative weight scalar which is used to balance the contribution for quaternion rank of $i$-th singular value. Note that the above three nonconvex rank surrogates are based on the laplace, geman, and weighted Schatten-$\gamma$ functions as defined in Table I, respectively.

Here, we first omit to give the closed-form solution of (9) and design an experiment on synthetic data to illustrate the superiority of the general model in Eq. (9) compared with the QNNM model in Eq. (7). We first use Matlab script randn$(m, r, 3)$ and randn$(r, n, 3)$ to generate two tensors $\mathcal{A} \in \mathbb{R}^{m\times r\times 3}$ and $\mathcal{B} \in \mathbb{R}^{r\times n\times 3}$. Then, the low-rank tensor $\mathcal{X} \in \mathbb{R}^{m\times n\times 3}$ is as the product of $\mathcal{A}$ and $\mathcal{B}$,

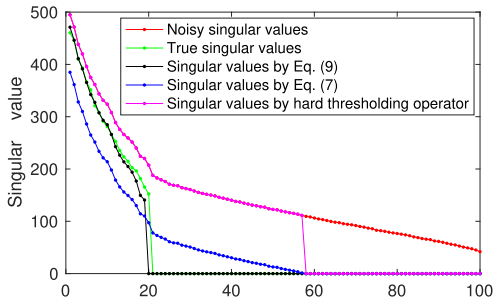


Fig. 1. Comparison of QNNM in Eq. (7) and the general model in Eq. (9). It is easy to observe that the singular values obtained by QNNM are overshrunk since each singular value is regularized equally. While the singular values estimated by the general model is closer to the original singular values.

*i.e.*, $\mathcal{X} = \mathcal{A} \otimes \mathcal{B}$[1]. And the additive Gaussian noise with zero mean and standard variance $\tau = 5$ is added into $\mathcal{X}$ to generate the noisy observation $\mathcal{Y}$, that is, $\mathcal{Y} = \mathcal{X} + \tau * \text{randn}(m, n, 3)$. Finally, we generate quaternion matrices $\dot{\mathbf{X}} \in \mathbb{H}^{m\times n}$ and $\dot{\mathbf{Y}} \in \mathbb{H}^{m\times n}$. Due to the fact that the distribution of singular values can represent the low-rank property [1], [3], [23], [40], we plot the singular values of clean quaternion matrix $\dot{\mathbf{X}}$, noisy observation $\dot{\mathbf{Y}}$, and obtained quaternion matrices by hard thresholding operator, models (7) and (9) in Fig. 1. In Fig. 1, $x$-axis denotes the number of singular values while $y$-axis represents the corresponding singular values. We can observe that the distribution of singular values of noisy observation $\dot{\mathbf{Y}}$ (the red curve) deviates significantly that of clean quaternion matrix $\dot{\mathbf{X}}$ (the green curve) due to the existence of Gaussian noise. Compared with the general model (the black curve), QNNM overshrinks each singular value since each singular value is thresholded with a same amount. While, the singular values obtained by the general model is closer to the original singular values. Meanwhile, there exits a rapid decline trend of original singular values of $\dot{\mathbf{X}}$, indicating that the quaternion matrix $\dot{\mathbf{X}}$ still has the low-rank characteristic. These observations can also verify the reasonableness of (9).

The remaining question is how to efficiently solve model (9). Actually, it is intractable to solve the general model in Eq. (9) due to the complicated operations of the quaternion algebra and the nonconvexity of $\phi(\cdot)$. Here, we focus on solving model (9) in complex domain due to the striking fact that computing SVD of the complex adjoint matrix is equivalent to computing QSVD of the corresponding quaternion matrix and the lower computation [22]. Based on *Definition 1* in Eqs (3) and (4), the quaternion matrix $\dot{\mathbf{Y}}$ can be conveniently and efficiently converted into the equivalent complex matrix $\chi_{\dot{\mathbf{Y}}}$, then the QSVD of $\dot{\mathbf{Y}}$ can be obtained by

$$\dot{\mathbf{U}} = col_o(\mathbf{U}_1) + col_o(-\overline{\mathbf{U}}_2)\mathbf{j}, \tag{13}$$

$$\Sigma = row_o(col_o(\Sigma')), \tag{14}$$

$$\dot{\mathbf{V}} = col_o(\mathbf{V}_1) + col_o(-\overline{\mathbf{V}}_2)\mathbf{j}, \tag{15}$$

where $\chi_{\dot{\mathbf{Y}}} = \mathbf{U}\Sigma'\mathbf{V}^{\triangleleft}$ is the SVD of complex matrix $\chi_{\dot{\mathbf{Y}}}$; $\mathbf{U} = [\mathbf{U}_1; \mathbf{U}_2] \in \mathbb{C}^{2m\times 2m}$, $\mathbf{V} = [\mathbf{V}_1; \mathbf{V}_2] \in \mathbb{C}^{2n\times 2n}$; $row_o(\mathbf{U})$ and $col_o(\mathbf{U})$ denote to extract the odd rows and odd columns of complex matrix $\mathbf{U}$, respectively. We first give the following theorem and then derive the solution of Eq. (9).

[1]Here, $\otimes$ is the tensor-tensor product, more details can be found in [50].

**Algorithm 1** Solving LRQA in Eq. (9) via DC

**Input:** Quaternion matrix $\dot{\mathbf{Y}}$, parameter $\lambda$.
**Initialize:** $t = 0,\ \sigma^{(0)} = 0$.
1: **while** not converged **do**
2: Compute the gradient of $\phi(\cdot)$;
3: Update $\sigma^{t+1}$ by Eq. (18);
4: **end while**
**Output:** $\dot{\mathbf{X}}^* = \dot{\mathbf{U}} diag(\sigma^*)\dot{\mathbf{V}}^{\triangleleft}$.

*Theorem 3:* Suppose $\dot{\mathbf{Y}} = \dot{\mathbf{U}}\Sigma_{\dot{\mathbf{Y}}}\dot{\mathbf{V}}^{\triangleleft}$ is the QSVD of $\dot{\mathbf{Y}} \in \mathbb{H}^{m\times n}$ and $\Sigma_{\dot{\mathbf{Y}}} = diag(\sigma_{\dot{\mathbf{Y}}})$, then the optional solution of Eq. (9) is $\dot{\mathbf{X}}^* = \dot{\mathbf{U}}\Sigma_{\dot{\mathbf{X}}}\dot{\mathbf{V}}^{\triangleleft}$, where $\Sigma_{\dot{\mathbf{X}}} = diag(\sigma^*)$ and $\sigma^*$ is obtained by

$$\sigma^* = \underset{\sigma\geq 0}{\text{argmin}} \frac{1}{2}\|\sigma - \sigma_{\dot{\mathbf{Y}}}\|_2^2 + \lambda * \phi(\sigma, \gamma). \tag{16}$$

The proof of *Theorem* 3 can be found in the supplementary material Appendix B. Based on *Theorem 3*, solving Eq. (9) is transformed into solving Eq. (16) in real domain. However, $\phi(\cdot)$ is continuous, concave, smooth, differentiable, and monotonically increasing on $[0, +\infty)$. Following [3], [17], the difference of convex (DC) method [51] is exploited to solve Eq. (16). The main idea of DC is to iteratively optimize it by linearizing the concave function $\phi(\cdot)$ and preserving the $l_2$-norm term, resulting in the following formula

$$\sigma^{t+1} = \underset{\sigma\geq 0}{\text{argmin}} \frac{1}{2}\|\sigma - \sigma_{\dot{\mathbf{Y}}}\|_2^2 + \lambda * \partial\phi(\sigma^t)^T\sigma, \tag{17}$$

where the superscript $t$ denotes the number of iterations; $\partial\phi(\sigma^t)$ is the gradient of $\phi(\cdot)$ at $\sigma^t$. The closed-form solution of Eq. (17) is

$$\sigma^{t+1} = \max\{\sigma_{\dot{\mathbf{Y}}} - \lambda * \partial\phi(\sigma^t), 0\}. \tag{18}$$

After several iterations, DC methods can converge to a local optimal $\sigma^*$. Finally, the optimal solution of Eq. (9) is $\dot{\mathbf{X}}^* = \dot{\mathbf{U}} diag(\sigma^*)\dot{\mathbf{V}}^{\triangleleft}$. The whole procedure of solving Eq. (9) is outlined in Algorithm 1.

## V. LRQA for Color Image Denoising and Inpainting

In this section, we apply the proposed LRQA to two color image processing tasks: color image denoising and inpainting.

### A. LRQA for Color Image Denoising

Image denoising is an ancient yet hot image processing topic and its aim is to remove unwanted noise $n$ and restore the clean image $x$ from the noisy observation $y$, which can be formulated as

$$y = x + n, \tag{19}$$

where $n$ is generally assumed to be additive white Gaussian noise with zero mean and variance $\tau_n^2$. However, this inverse

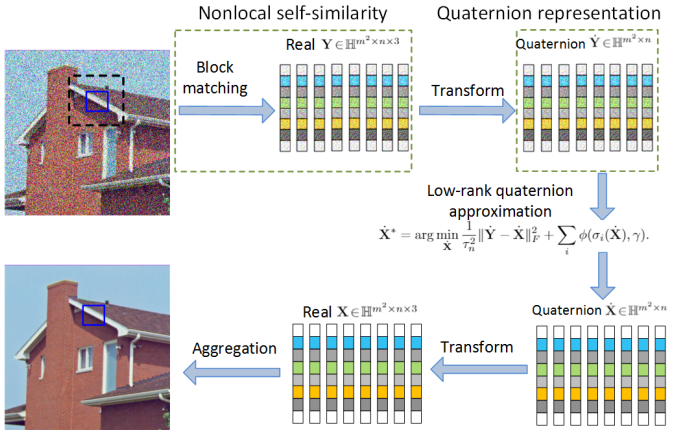


Fig. 2. The flowchart of color image denoising by LRQA.

problem (19) is severely ill-posed without any priors of image $x$. Therefore, regularization for image prior knowledge is needed to stabilize the inversion process. Here, we consider the nonlocal self-similarity (NSS) and low-rank property for image denoising task.

Following the procedure of NSS, we first divide the noisy color image $y$ into many overlapping color patches of size $m\times m\times 3$ (for example, $6\times 6\times 3$). Instead of modeling color image pixel as a scalar in traditional sparse representation and LRMA-based methods [1], [2], we use a pure quaternion number to represent a color image pixel. This is also the main difference with these existing denoising methods [1], [2], [17]. For an exemplar patch of size $m\times m\times 3$ at position $i$ denoted by $\dot{y}_i\in\mathbb{H}^{m^2}$ ($\dot{y}_i=y_r\mathbf{i}+y_g\mathbf{j}+y_b\mathbf{k}$, where $y_r,\ y_g,\ y_b\in\mathbb{R}^{m^2}$ are the corresponding patch vectors of R, G, B channels), its similar patch group consists of $n$ nearest neighbor patches within its local searching window (where $n$ includes $\dot{y}_i$) and then is stacked as quaternion column vectors of quaternion matrix $\dot{\mathbf{Y}}_i=[\dot{y}_{i1},\ \dot{y}_{i2},\cdots,\ \dot{y}_{in}]\in\mathbb{H}^{m^2\times n}$. The above whole procedure is depicted in Fig 2. The key assumption of NSS is that for any exemplar patch $y_i$, there exist many similar local patches across a natural image, which implies that the constructed matrix $\mathbf{Y}_i$ is low-rank [1]. Similarly, we assume that our constructed quaternion matrix $\dot{\mathbf{Y}}_i$ also has the low-rank characteristic in the quaternion domain. Based on this assumption, the clean patch group $\dot{X}_i$ can be obtained by

$$\min_{\dot{X}}\frac{1}{\tau_n^2}\|\dot{\mathbf{Y}}-\dot{\mathbf{X}}\|_F^2+\sum_i\phi\big(\sigma_i(\dot{\mathbf{X}}),\gamma\big),\tag{20}$$

where $\|\dot{\mathbf{Y}}-\dot{\mathbf{X}}\|_F^2$ of Eq. (20) is the data fidelity term regularized by the Frobenius-norm, which is consistent with the assumption in (19) because $l_2$-norm is optimal to suppressing Gaussian noise. The optimal solution of Eq. (20) can be obtained by Algorithm 1.

Once the clean patch group $\dot{\mathbf{X}}$ is learned by Eq. (20), we transform quaternion matrix $\dot{\mathbf{X}}$ into real matrix $\mathbf{X}$ and aggregate all patches together to form the clean color image $x$. Following [1], [2], [23], we adopt the iterative regularization scheme [54] as follows

$$y^{(k)}=\hat{x}^{(k-1)}+\delta(y-\hat{x}^{(k-1)}),\tag{21}$$

where $k$ and $\delta$ are the iteration number and the relaxation parameter, respectively. The flowchart of LRQA for color image denoising is shown in Fig. 2. The whole procedure of color image denoising is summarized in Algorithm 2.

**Algorithm 2** Color image denoising by LRQA in Eq. (20)

**Input:** Color noisy image $y$, noise level $\tau$, relaxation parameter $\delta$.
**Initialize:** $\hat{x}^{(0)}=y,\ y^{(0)}=y$.
**for** $k=1:K$ **do**
  Iterative regularization $y^{(k)}=\hat{x}^{(k-1)}+\delta(y-\hat{x}^{(k-1)})$;
  **for** each exemplar patch $y_i$ in $y^{(k)}$ **do**
    Construct similar patch group $\mathbf{Y}_i$;
    Transform real matrix $\mathbf{Y}_i$ into quaternion matrix $\dot{\mathbf{Y}}_i$;
    Obtain the estimator $\dot{\mathbf{X}}_i$ by Eq. (20) (Algorithm 1);
    Transform quaternion matrix $\dot{\mathbf{X}}_i$ into real matrix $\mathbf{X}_i$;
  **end for**
  Aggregate $\{\mathbf{X}_i\}$ to form the clean image $\hat{x}^{(k)}$;
**end for**
**Output:** Denoised image $\hat{x}^{(K)}$.

### *B. LRQA for Color Image Inpainting*

In this section, we further apply the proposed LRQA to color image inpainting. Image inpainting targets at filling the missing pixels from the incomplete observation $\mathbf{D}$. As stated in [4], [13], and [16], the real images can be approximately recovered by a low-rank matrix. Therefore, we use the proposed LRQA to recover the color image by the following model:

$$\begin{aligned}&\min_{\dot{\mathbf{X}}}\sum_i\phi\big(\sigma_i(\dot{\mathbf{X}}),\gamma\big)\\&s.t.\ \ \mathcal{P}_\Omega(\dot{\mathbf{X}})=\mathcal{P}_\Omega(\dot{\mathbf{D}}),\end{aligned}\tag{22}$$

where $\Omega$ is the set of observed entries of $\mathbf{D}$. $\mathcal{P}_\Omega(\dot{\mathbf{X}})$ denotes the same size quaternion matrix as $\dot{\mathbf{X}}$ with $\mathcal{P}_\Omega(\dot{\mathbf{X}})_{i,j}=\dot{\mathbf{D}}_{i,j}$ if $(i,j)\in\Omega$ and zero otherwise.

To make $\dot{\mathbf{X}}$ separable, we use the variable-splitting technique [55] and introduce one auxiliary quaternion variable $\dot{\mathbf{E}}$ into Eq. (22) yielding the following model:

$$\begin{aligned}&\min_{\dot{\mathbf{X}},\dot{\mathbf{E}}}\sum_i\phi\big(\sigma_i(\dot{\mathbf{X}}),\gamma\big)\\&s.t.\ \ \dot{\mathbf{X}}+\dot{\mathbf{E}}=\dot{\mathbf{D}},\quad\mathcal{P}_\Omega(\dot{\mathbf{E}})=\dot{\mathbf{0}}.\end{aligned}\tag{23}$$

The partial augmented Lagrangian function of Eq. (23) is defined as

$$\mathcal{L}_\mu(\dot{\mathbf{X}},\dot{\mathbf{E}};\dot{\Theta})=\sum_i\phi\Big(\sigma_i(\dot{\mathbf{X}}),\gamma\Big)+\frac{\mu}{2}\|\dot{\mathbf{D}}-\dot{\mathbf{X}}-\dot{\mathbf{E}}+\frac{\dot{\Theta}}{\mu}\|_F^2,\tag{24}$$

where $\mu>0$ is the penalty parameter. $\dot{\Theta}$ is the Lagrangian multiplier.

**Algorithm 3** Color image inpainting by LRQA in Eq. (22)

**Input:** Incomplete color image $D$, the set of observed entries $\Omega$.

**Initialize:** $\dot{\mathbf{X}}^{(0)} = \dot{\mathbf{E}}^{(0)} = \dot{0},\ \Theta^{(0)} = \dot{0},\ \mu^0 = 1e^{-3},\ \beta = 1.1.$

1: **while** not converged **do**
2: Update $\dot{\mathbf{X}}^{k+1}$ by Eq. (25) (Algorithm 1);
3: Update $\dot{\mathbf{E}}^{k+1}$ by Eq. (26);
4: Update $\dot{\Theta}^{k+1}$ and $\mu^{k+1}$ by Eqs. (27) and (28);
5: **end while**

**Output:** $\dot{\mathbf{X}}$.

Under the complex altering direction method of multipliers framework [56], we can iteratively update all variables as follows

$$\dot{\mathbf{X}}^{k+1} = \underset{\dot{\mathbf{X}}}{\operatorname{argmin}} \sum_i \phi\Big(\sigma_i(\dot{\mathbf{X}}), \gamma\Big) + \frac{\mu^k}{2}\|\dot{\mathbf{D}} - \dot{\mathbf{X}} - \dot{\mathbf{E}}^k + \frac{\dot{\Theta}^k}{\mu^k}\|_F^2, \tag{25}$$

$$\dot{\mathbf{E}}^{k+1} = \underset{\mathcal{P}_\Omega(\dot{\mathbf{E}})=\dot{\mathbf{0}}}{\operatorname{argmin}} \|\dot{\mathbf{D}} - \dot{\mathbf{X}}^{k+1} - \dot{\mathbf{E}} + \frac{\dot{\Theta}^k}{\mu^k}\|_F^2, \tag{26}$$

$$\dot{\Theta}^{k+1} = \dot{\Theta}^k + \mu^k(\dot{\mathbf{D}} - \dot{\mathbf{X}}^{k+1} - \dot{\mathbf{E}}^{k+1}), \tag{27}$$

$$\mu^{k+1} = \min\{\beta * \mu^k, \mu_{max}\}, \tag{28}$$

where $0 < \beta < \frac{1+\sqrt{5}}{2}$. $\mu_{max}$ is the maximum value of the penalty parameter $\mu$. The closed-form solution of Eq. (25) can be obtained by Algorithm 1. The optimal $\dot{\mathbf{E}}^{k+1}$ of Eq. (26) is $\dot{\mathbf{D}} - \dot{\mathbf{X}}^{k+1} + \frac{\dot{\Theta}^k}{\mu^k}$. To guarantee the observed entries unchanged, $\mathcal{P}_\Omega(\dot{\mathbf{E}}) = \dot{\mathbf{0}}$ is held throughout the iteration. The whole procedure of LRQA for color image inpainting is summarized in Algorithm 3, in which the stopping criterion is defined as follows

$$\frac{\|\dot{\mathbf{D}} - \dot{\mathbf{X}}^{k+1} - \dot{\mathbf{E}}^{k+1}\|_F^2}{\|\dot{\mathbf{D}}\|_F^2} \le tol, \tag{29}$$

where $tol > 0$ is a pre-defined tolerance.

## VI. Experiments Results

In this section, we conduct several experiments to verify the effectiveness of the proposed LRQA including two real color image processing tasks: denoising and inpainting. All experiments are run in Matlab R2016a on a 64-bit workstation with a 1.90GHz CPU and 16GB memory.

### A. Simulated Noisy Color Image Denoising

**Data:** We select 12 widely used color images as shown in Fig. 4. To generate simulated noisy color images, the additive white Gaussian noise with zero mean and variance $\tau_n^2$ is added to these clean color images. Three noise levels $\tau = 10,\ 30,\ 50$ are considered in our experiments.

**Compared methods:** We compare the proposed LRQA with several state-of-the-art SR and LRMA-based denoising methods, including LSCD [52], KQSVD [22], NNM [16], and WNNM [53]. For a comprehensive comparison, we implement WNNM with two versions by the concatenation model and the monochromatic model, denoted as WNNM-1 and WNNM-2, respectively. Note that KQSVD is recently proposed color image denoising method based on the vector sparse representation model and quaternion algebra. LRQA using the nuclear norm, Laplace function, Geman function, and weighted Schatten norm are denoted as LRQA-1, LRQA-2, LRQA-3, and LRQA-4, respectively.

**Parameter setting:** In Algorithm 2, when $\tau = 10,\ 30,\ 50$, we set patch size to $6\times6$, $7\times7$, $8\times8$, respectively. The number of similar patches group is set to 70, 90, and 120, respectively. The iteration $K$ is 8, 12, and 14, respectively. In all cases, the iterative relaxation parameter $\delta$ is fixed to 0.1. Except for parameter $\gamma$ in LRQA-4, there exists another parameter $\omega$ which is set as

$$\omega_i^k = c\sqrt{n}/(\sigma_i(\dot{\mathbf{X}}^{k-1}) + \epsilon),$$

where $c = 3 * \sqrt{2}$; $n$ is the number of similar patches group; $\sigma_i(\dot{\mathbf{X}}_i^{k-1})$ is the $i$-th singular value of $\dot{\mathbf{X}}^{k-1}$; $\epsilon$ is a small value to avoid being divided by zero.

**Evaluation measures:** The quantitative quality indexes, *i.e.*, Peak Signal-to-Noise Ration (PSNR) and Structure Similarity (SSIM) [57], are selected to measure the quality of all denoising methods. In general, the higher values the PSNR and SSIM are, the better the denoising quality is. All competing denoising methods are from the source codes and follow the parameter settings in the orginal papers.

The PSNR and SSIM values of all denoising methods are reported in Table II. The highest PSNR and SSIM values are highlighted in bold, and the second-highest ones are underlined. The average PSNR values of all denoising methods are also computed. From this table, we have the following observations:

- In most cases, the proposed LRQA yields the better denoising results than the best competing method WNNM. Among them, LRQA-4 (LRQA using the weighted Schatten $\gamma$-norm as defined in Eq. (12)) achieved the highest PSNR and SSIM values. The average improvement of the proposed LRQA-4 over WNNM is around 0.113dB, 0.323dB, and 0.366dB when $\tau = 10,\ 30,\ 50$, respectively. Besides, when the variance of Gaussian noise increases, the advantage of LRQA over WNNM is more obvious. These observations verify the effectiveness of the low-rank quaternion approximation over the low-rank matrix approximation in real domain for color image denoising. The main reason is that the proposed LRQA models tree color channels into three imaginary parts such that the whole color image can be processed in a holistic manner and more details of color image can be preserved.
- LRQA-1 (LRQA using the quaternion nuclear norm) performs worse than our proposed LRQA variants in all cases. This is because that the quaternion nuclear norm shrinks each singular value equally, resulting in over-penalizing larger singular values. This observation is consistent with that in Fig. 1.
- WNNM-1 outperforms WNNM-2 in all cases, in which WNNM-1 concatenates three color channels, while the second one restores each channel of a color image

TABLE II
QUANTITATIVE EVALUATION (PSND/SSIM) OF THE DIFFERENT IMAGE DENOISING ALGORITHMS

| | $\tau = 10$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Image | LSCD [52] | KQSVD [22] | NNM [16] | WNNM-1 [53] | WNNM-2 [53] | LRQA-1 | LRQA-2 | LRQA-3 | LRQA-4 |
| | | | | | | | $\gamma = 0.15$ | $\gamma = 0.18$ | $\gamma = 1.14$ |
| Img1 | 34.47/0.934 | 33.81/0.927 | 33.82/0.917 | 34.66/0.938 | 33.86/0.917 | 33.11/0.867 | 34.73/0.939 | 34.73/0.939 | **34.77/0.941** |
| Img2 | 28.79/0.870 | 28.47/0.877 | 29.85/0.896 | 30.45/0.907 | 30.19/0.902 | 30.14/0.890 | **30.48/0.908** | 30.45/0.908 | **30.48/0.908** |
| Img3 | 32.73/0.906 | 32.78/0.913 | 33.78/0.929 | 34.78/0.945 | 33.51/0.920 | 33.13/0.902 | 34.92/0.946 | 34.91/0.946 | **34.99**/0.948 |
| Img4 | 31.08/0.922 | 30.90/0.918 | 32.12/0.931 | 32.95/0.940 | 32.05/0.934 | 31.98/0.929 | 33.08/0.944 | 33.08/0.944 | **33.11/0.944** |
| Img5 | 33.97/0.862 | 34.20/0.846 | 35.25/0.888 | 35.75/0.897 | 34.23/0.878 | 34.01/0.847 | 35.90/0.905 | 35.89/0.905 | **36.03/0.907** |
| Img6 | 34.73/0.928 | 34.20/0.922 | 34.36/0.918 | 35.23/0.934 | 33.92/0.902 | 33.47/0.876 | 35.30/0.934 | 35.30/0.934 | **35.41/0.939** |
| Img7 | 33.16/0.944 | 33.12/0.942 | 33.65/0.944 | 34.67/0.958 | 33.33/0.927 | 32.96/0.913 | 34.70/0.957 | 34.70/0.957 | **34.75/0.959** |
| Img8 | 31.98/0.916 | 31.88/0.916 | 32.07/0.911 | **32.77**/0.922 | 31.91/0.898 | 31.80/0.882 | 32.73/0.924 | 32.72/0.924 | 32.76/0.925 |
| Img9 | 33.47/0.877 | 33.61/0.884 | 33.56/0.880 | 34.05/0.888 | 32.89/0.858 | 32.84/0.843 | 34.07/0.887 | 34.14/0.891 | **34.16/0.891** |
| Img10 | 33.76/0.974 | 32.56/0.965 | 33.71/0.973 | **35.05/0.984** | 33.37/0.967 | 32.91/0.950 | 34.97/0.984 | 34.96/0.984 | 34.96/0.984 |
| Img11 | 34.24/0.940 | 33.34/0.931 | 33.33/0.925 | 34.37/0.943 | 33.42/0.924 | 32.86/0.897 | 34.55/0.946 | 34.56/0.946 | **34.57/0.947** |
| Img12 | 32.60/0.929 | 31.11/0.917 | 33.01/0.925 | 33.93/0.940 | 32.89/0.916 | 32.53/0.893 | 33.99/0.941 | 34.01/0.941 | **34.02/0.943** |
| Aver. | 32.915 | 32.498 | 33.209 | 34.055 | 33.048 | 32.645 | 34.118 | 34.121 | **34.168** |
| | $\tau = 30$ | | | | | | | | |
| Image | | | | | | | $\gamma = 0.4$ | $\gamma = 0.5$ | $\gamma = 1.11$ |
| Img1 | 28.33/0.813 | 28.67/0.845 | 27.71/0.825 | 28.97/0.857 | 27.03/0.782 | 28.56/0.837 | 29.08/0.858 | 29.19/0.866 | **29.27/0.867** |
| Img2 | 24.13/0.663 | 24.17/0.673 | 23.75/0.621 | 24.47/0.679 | 23.50/0.687 | 24.34/0.671 | 24.72/0.708 | 24.67/0.702 | **24.80/0.712** |
| Img3 | 27.88/0.792 | 27.91/0.799 | 27.72/0.804 | 28.91/0.836 | 26.68/0.757 | 28.68/0.830 | 29.02/0.840 | 29.27/0.852 | **29.40/0.854** |
| Img4 | 25.92/0.774 | 25.83/0.748 | 25.53/0.733 | 26.94/0.793 | 25.24/0.767 | 26.68/0.793 | 27.26/0.819 | 27.38/0.819 | **27.51/0.824** |
| Img5 | 29.77/0.772 | 29.83/0.787 | 31.05/0.804 | 31.91/0.818 | 28.52/0.727 | 31.52/0.806 | 31.69/0.815 | 32.00/0.821 | **32.18/0.826** |
| Img6 | 29.40/0.831 | 29.19/0.825 | 28.64/0.821 | 29.76/0.848 | 27.60/0.764 | 29.47/0.833 | 29.80/0.842 | 30.06/0.856 | **30.16/0.859** |
| Img7 | 27.78/0.854 | 27.70/0.859 | 27.31/0.862 | 28.71/0.888 | 26.30/0.791 | 28.18/0.875 | 28.72/0.880 | 28.92/0.893 | **29.01/0.895** |
| Img8 | 26.73/0.796 | 26.95/0.810 | 25.82/0.779 | 26.93/0.813 | 25.23/0.741 | 26.35/0.798 | 26.95/0.812 | 26.97/0.820 | **27.05/0.823** |
| Img9 | 28.83/0.739 | 28.90/0.743 | 28.21/0.735 | 29.15/0.761 | 26.96/0.666 | 28.79/0.746 | 28.99/0.743 | 29.25/0.766 | **29.35/0.770** |
| Img10 | 27.61/0.913 | 26.83/0.911 | 26.65/0.921 | 28.45/0.943 | 25.87/0.873 | 27.53/0.928 | 28.43/0.942 | 28.62/0.946 | **28.63/0.946** |
| Img11 | 28.51/0.830 | 27.95/0.809 | 27.13/0.793 | 28.32/0.831 | 26.52/0.767 | 28.05/0.819 | 28.59/0.838 | 28.73/0.846 | **28.82/0.848** |
| Img12 | 27.41/0.824 | 26.70/0.811 | 26.41/0.813 | 28.08/0.847 | 26.04/0.763 | 27.34/0.829 | 28.06/0.844 | 28.24/0.854 | **28.29/0.855** |
| Aver. | 27.692 | 27.553 | 27.161 | 28.383 | 26.291 | 27.958 | 28.443 | 28.608 | **28.706** |
| | $\tau = 50$ | | | | | | | | |
| Image | | | | | | | $\gamma = 0.5$ | $\gamma = 0.65$ | $\gamma = 1.1$ |
| Img1 | 21.57/0.391 | 25.96/0.773 | 25.05/0.755 | 26.56/0.801 | 24.71/0.724 | 26.09/0.790 | 26.73/0.809 | 26.73/0.814 | **26.89/0.813** |
| Img2 | 20.49/0.437 | 22.39/0.511 | 21.93/0.462 | 22.61/0.532 | 21.72/0.573 | 22.48/0.519 | 22.79/0.573 | 22.73/0.553 | **22.93/0.573** |
| Img3 | 21.83/0.462 | 25.50/0.712 | 24.88/0.710 | 26.40/0.762 | 24.33/0.666 | 26.09/0.756 | 26.80/0.781 | 26.80/0.781 | **26.89/0.781** |
| Img4 | 20.95/0.517 | 23.44/0.592 | 22.72/0.549 | 24.30/0.660 | 23.10/0.671 | 23.94/0.642 | 24.75/0.703 | 24.73/0.691 | **24.94/0.710** |
| Img5 | 23.00/0.402 | 27.32/0.732 | 28.38/0.765 | 29.95/0.789 | 26.21/0.677 | 29.46/0.783 | 30.18/0.787 | 30.19/0.798 | **30.26/0.794** |
| Img6 | 22.82/0.462 | 26.63/0.748 | 26.17/0.752 | 27.44/0.788 | 25.41/0.694 | 27.21/0.784 | 27.81/0.796 | 27.80/0.807 | **27.90/0.802** |
| Img7 | 21.62/0.539 | 25.06/0.784 | 24.14/0.781 | 26.09/0.831 | 23.83/0.716 | 25.39/0.820 | 26.29/0.835 | 26.29/0.843 | **26.42/0.837** |
| Img8 | 21.28/0.459 | 24.49/0.723 | 22.88/0.674 | 24.62/0.739 | 23.06/0.674 | 23.90/0.721 | 24.64/0.745 | 24.66/0.748 | **24.76/0.750** |
| Img9 | 22.99/0.397 | 26.63/0.653 | 25.97/0.652 | 27.04/0.688 | 25.04/0.591 | 26.66/0.678 | 27.10/0.688 | 27.15/0.697 | **27.25/0.695** |
| Img10 | 21.05/0.680 | 23.87/0.854 | 22.96/0.852 | 25.32/0.900 | 23.19/0.813 | 24.34/0.884 | 25.51/0.907 | 25.51/0.907 | **25.64/0.905** |
| Img11 | 22.27/0.495 | 25.23/0.701 | 24.45/0.686 | 25.84/0.746 | 24.26/0.681 | 25.57/0.737 | 26.28/0.769 | 26.28/0.769 | **26.39/0.770** |
| Img12 | 21.86/0.503 | 24.54/0.733 | 23.17/0.719 | 25.31/0.778 | 23.53/0.683 | 24.57/0.761 | 25.46/0.787 | 25.47/0.788 | **25.60/0.785** |
| Aver. | 21.811 | 25.088 | 24.392 | 25.957 | 24.033 | 25.475 | 26.195 | 26.195 | **26.323** |

**Bold** fonts denote the best performance; underlined ones represent the second-best results in all tables.

TABLE III
AVERAGE IMPROVEMENT (PSNR(DB)) OF LRQA OVER KQSVD [22]

| | LRQA-1 | LRQA-2 | LRQA-3 | LRQA-4 |
|---|---|---|---|---|
| $\tau = 10$ | 0.147 | 1.620 | 1.623 | 1.670 |
| $\tau = 30$ | 0.405 | 0.890 | 1.055 | 1.153 |
| $\tau = 50$ | 0.387 | 1.107 | 1.107 | 1.235 |

individually and then combined all results together. This indicates that the concatenation model-based denoising method may capture more image information than the monochromatic model-based ones.

We also give the visual comparison between all competing methods and LRQA as shown in Figs. 3 and 5. Obviously, some Gaussian noise still remains in the denoised images by LSCD. KQSVD and NNM can remove Gaussian noise to some extent, but oversmooth the results as shown in the highlighted red rectangle in Figs. 3 and 5. The proposed LRQA-4 achieves the best visual performance among all denoising methods.

To compare the proposed LRQA with the quaternion-based denoising method: KQSVD [22], we also give the average improvement of LRQA over KQSVD as shown in Table III and their average runtime of them. The average runtime of KQSVD on all color images is 112.35$s$ while these of the proposed LRQA-1, LRQA-2, LRQA-3 and LRQA-4 are 1988.12$s$, 2112.32$s$, 2123.25$s$, and 2205.22$s$, respectively. One can see that KQSVD is running faster than the proposed LRQA. The main computation burden of LRQA is performing the quaternion singular value decomposition and nonlocal self-similarity. It is natural because the proposed LRQA takes the low-rank property and the nonlocal self-similarity prior into consideration. However, all proposed LRQA methods achieve higher PSNR values over KQSVD. Specifically, LRQA-3 and LRQA-4 improve more than 1.0dB.

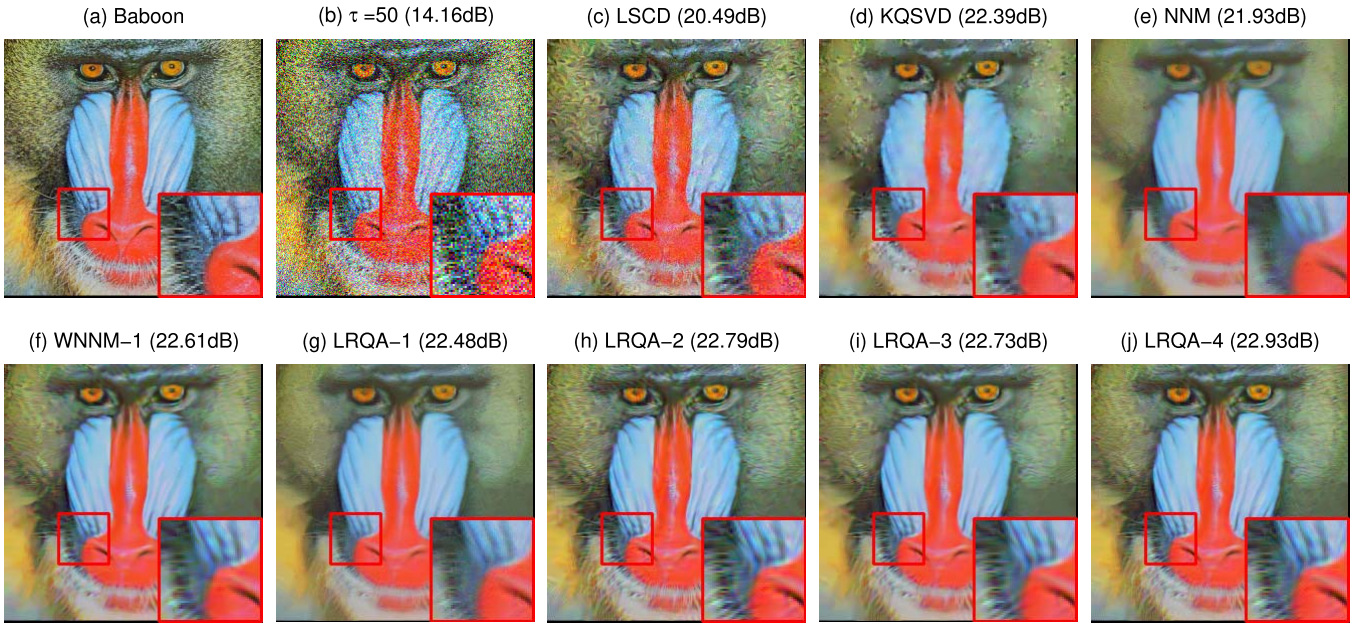


Fig. 3. Color image denoising results on "Baboon." (a) Original image; (b) Noisy image corrupted by Gaussian noise with variance $\tau = 50$, respectively; The denoised image reconstructed by: (c) LSCD [52], (d) KQSVD [22], (e) NNM [16], (f) WNNM-1 [53], (g) QLRMA-1, (h) QLRMA-2, (i) QLRMA-3, (j) QLRMA-4. The figure is viewed better in zoomed PDF.

Fig. 4. The 12 color images of size $256 \times 256 \times 3$.

### B. Real Noisy Color Image Denoising

In order to give an overall evaluation of LRQA, in this section, we use two real noisy color images[2], *i.e.*, Dog and Bears, to test the proposed LRQA for real noise removal. These two images are mainly contaminated by the Gaussian noise [11]. Following [1], we first use the method in [59] to estimate the noise level, since there is no groundtruth. We select WNNM [53], NC [60], TLR_HTV [11] as the competing methods. Fig. 6 displays the denoising results of all methods. We can observe that some noise also left in the images by NC and TLR_HTV may oversmooth the image. The images by WNNM and LRQA are visually similar. While the structural information among three channels and image details are well captured by LRQA. This further verify the advantage and effectiveness of the proposed LRQA.

### C. Color Image Inpainting

In this section, we test the performance of the proposed LRQA for color image inpainting. In particular, we just consider the model in Eq. (22) using the quaternion nuclear norm, laplace function and geman function, corresponding to LRQA-1, LRQA-2, and LRQA-3, respectively, since LRQA using other nonconvex functions has similar inpainting performance. We compare LRQA with several representative LRMA-based image inpainting methods, including NNM [16], TNNR [4], WNNM, MC-NC [21], LMaFit [58], D-N [13] and F-N [13]. Among them, the first four methods belong to the matrix rank minimization-based LRMA ones, while the others are the matrix factorization-based LRMA ones. It is worth noting that D-N and F-N are recently proposed methods for image inpainting task. 8 color images are selected as the test samples as shown in Fig. 7. Following the experimental settings in [4], [13], all competing methods are performed on each channel of the test images individually. We also use the PSNR and SSIM as the quantitative quality.

The quantitative PSNR and SSIM values of all inpainting methods are reported in Table IV, in which $\rho$ represents the ratio of the missing pixels. In most cases, LRQA-2 and LRQA-3 outperform other competing methods. When the ratio of missing pixels $\rho = 0.50,\ 0.75$, the average improvement of LRQA-2 over the best competing method TNNR is around 0.353dB and 0.969dB, respectively. When $\rho = 0.85$, LRQA-2 achieves average 1.051dB improvement over F-N, further verifying the advantages of LRQA. MC-NC achieves the comparable results with the proposed LRQA-2 when $\rho = 0.50$, while it performs extremely worse than most of other methods when $\rho = 0.85$. The visual comparison between LRQA and all competing inpainting methods on the seventh images with 85% random missing pixels is shown in Fig. 8. It is obvious that the proposed LRQA achieves the best inpainting results. In particular, LMaFit needs to predefine the rank prior, which is difficult to obtain in real applications [11], and can not well restore the details.

[2]http://demo.ipol.im/demo/125/

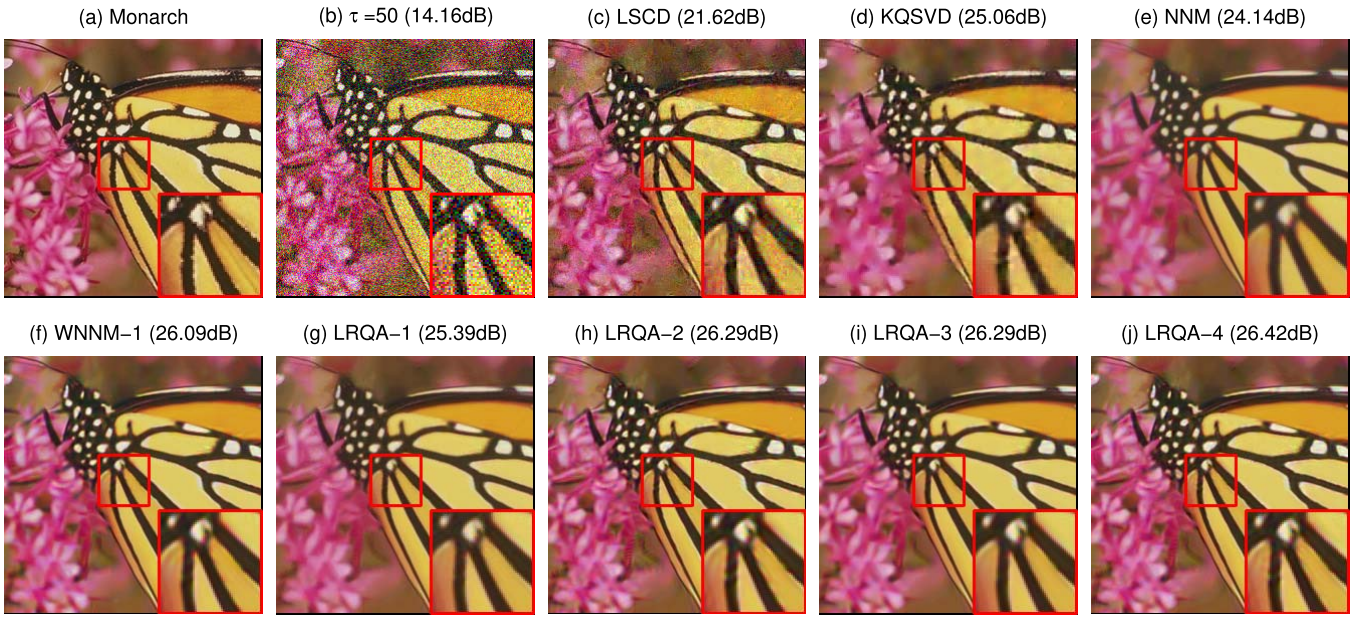


Fig. 5. Color image denoising results on "Monarch." (a) Original image; (b) Noisy image corrupted by Gaussian noise with variance $\tau = 50$; The denoised image reconstructed by: (c) LSCD [52], (d) KQSVD [22], (e) NNM [16], (f) WNNM-1 [53], (g) QLRMA-1, (h) QLRMA-2, (i) QLRMA-3, (j) QLRMA-4. The figure is viewed better in zoomed PDF.

TABLE IV
QUANTITATIVE EVALUATION (PSNR/SSIM) OF THE DIFFERENT IMAGE INPAINTING ALGORITHMS

| | $\rho = 0.50$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Image | LMaFit[58] | NNM[16] | TNNR[4] | WNNM[1] | MC-NC[21] | D-N[13] | F-N[13] | LRQA-1 | LRQA-2 | LRQA-3 |
| | | | | | | | | | $\gamma = 20$ | $\gamma = 20$ |
| Img1 | 21.33/0.779 | 32.83/0.912 | 33.49/0.921 | 31.54/0.875 | 32.71/0.908 | 27.33/0.799 | 27.28/0.800 | 32.78/0.912 | 33.50/0.922 | **33.61/0.923** |
| Img2 | 20.24/0.443 | 24.82/0.696 | 24.46/0.655 | 22.06/0.510 | 23.78/0.603 | 21.06/0.406 | 21.07/0.408 | 24.77/0.692 | **24.95/0.692** | 24.76/0.677 |
| Img3 | 22.14/0.644 | 27.36/0.801 | 27.53/0.810 | 25.02/0.707 | **27.65/0.820** | 23.09/0.526 | 23.09/0.526 | 27.33/0.801 | 27.58/0.810 | 27.53/0.808 |
| Img4 | 23.17/0.558 | 27.73/0.764 | 26.62/0.708 | 25.24/0.636 | 26.86/0.723 | 24.14/0.490 | 24.15/0.490 | 27.69/0.764 | **27.88/0.769** | 27.81/0.764 |
| Img5 | 24.07/0.941 | 28.02/0.963 | 28.42/0.966 | 26.96/0.948 | 28.63/0.967 | 21.40/0.864 | 21.40/0.863 | 28.00/0.963 | 29.03/0.969 | **29.14/0.970** |
| Img6 | 21.95/0.775 | 27.69/0.884 | 27.65/0.887 | 24.92/0.805 | **27.94/0.898** | 24.82/0.750 | 24.82/0.749 | 27.66/0.883 | 27.85/0.885 | 27.72/0.882 |
| Img7 | 28.01/0.864 | 35.91/0.953 | 36.61/0.956 | 33.96/0.905 | 36.20/0.950 | 30.63/0.814 | 30.59/0.814 | 35.91/0.954 | 36.60/0.958 | **36.81/0.959** |
| Img8 | 18.67/0.700 | 24.11/0.836 | 24.27/0.840 | 21.32/0.748 | 24.16/0.837 | 19.71/0.610 | 19.72/0.611 | 24.10/0.836 | **24.48/0.845** | 24.32/0.839 |
| Aver. | 22.446 | 28.559 | 28.632 | 26.376 | 28.490 | 24.022 | 24.015 | 28.531 | **28.985** | 28.964 |
| | $\rho = 0.75$ | | | | | | | | | |
| Image | | | | | | | | | $\gamma = 40$ | $\gamma = 40$ |
| Img1 | 17.05/0.424 | 23.51/0.504 | 28.30/0.789 | 27.00/0.681 | 28.52/0.791 | 26.44/0.735 | 26.36/0.739 | 27.64/0.767 | 28.64/0.799 | **28.66/0.797** |
| Img2 | 13.94/0.139 | 20.72/0.430 | 19.99/0.362 | 18.19/0.250 | 18.53/0.265 | 20.11/0.330 | 20.21/0.336 | 21.04/0.437 | **21.41/0.444** | 21.27/0.429 |
| Img3 | 15.76/0.219 | 22.27/0.496 | 23.06/0.562 | 21.09/0.404 | 22.70/0.521 | 22.27/0.467 | 22.38/0.472 | 23.40/0.562 | **23.72/0.578** | 23.51/0.566 |
| Img4 | 16.96/0.190 | 23.99/0.516 | 22.88/0.461 | 21.43/0.351 | 23.10/0.448 | 23.42/0.436 | 23.47/0.439 | 24.24/0.535 | **24.53/0.547** | 24.49/0.542 |
| Img5 | 15.55/0.674 | 19.99/0.829 | 23.05/0.894 | 21.26/0.843 | 23.30/0.900 | 20.57/0.837 | 20.55/0.836 | 22.15/0.878 | **23.57/0.901** | 23.51/0.899 |
| Img6 | 17.27/0.541 | 23.59/0.755 | 23.78/0.770 | 21.91/0.662 | 23.77/0.748 | 23.89/0.719 | 23.99/0.720 | 24.32/0.774 | **24.68/0.781** | 24.62/0.778 |
| Img7 | 19.28/0.497 | 30.30/0.852 | 30.08/0.848 | 28.78/0.744 | 31.19/0.856 | 29.37/0.782 | 29.12/0.779 | 30.88/0.863 | 31.78/0.877 | **31.96/0.879** |
| Img8 | 14.11/0.446 | 19.48/0.529 | 19.45/0.647 | 17.52/0.563 | 19.00/0.585 | 18.65/0.541 | 18.73/0.546 | 20.27/0.648 | **20.50/0.666** | 20.38/0.664 |
| Aver. | 16.211 | 22.981 | 23.824 | 22.149 | 23.766 | 23.091 | 23.102 | 24.241 | **24.793** | 24.704 |
| | $\rho = 0.85$ | | | | | | | | | |
| Image | | | | | | | | | $\gamma = 40$ | $\gamma = 40$ |
| Img1 | 11.24/0.043 | 12.85/0.104 | 24.52/0.619 | 23.72/0.495 | 20.83/0.312 | 24.83/0.648 | 24.74/0.658 | 24.95/0.647 | **26.14/0.683** | 26.06/0.672 |
| Img2 | 11.22/0.077 | 17.45/0.281 | 16.81/0.181 | 16.31/0.137 | 15.96/0.146 | 19.01/0.246 | 19.26/0.260 | 19.46/0.321 | 19.90/0.322 | **19.91/0.322** |
| Img3 | 10.91/0.085 | 16.26/0.251 | 20.07/0.363 | 18.81/0.245 | 18.03/0.226 | 21.09/0.383 | 21.35/0.398 | 21.67/0.432 | **22.03/0.449** | 21.91/0.438 |
| Img4 | 12.67/0.079 | 19.76/0.310 | 20.31/0.285 | 19.72/0.228 | 19.54/0.224 | 22.26/0.363 | 22.53/0.378 | 22.80/0.414 | **23.08/0.429** | 22.95/0.418 |
| Img5 | 10.08/0.301 | 14.31/0.568 | 18.97/0.777 | 16.43/0.653 | 16.69/0.671 | 19.28/0.789 | 19.36/0.793 | 19.04/0.788 | **20.42/0.824** | 20.32/0.820 |
| Img6 | 12.85/0.263 | 17.50/0.511 | 21.17/0.652 | 20.14/0.570 | 19.30/0.537 | 22.66/0.674 | 22.85/0.682 | 22.82/0.707 | **23.34/0.714** | 23.23/0.710 |
| Img7 | 15.21/0.193 | 25.49/0.648 | 26.32/0.694 | 26.19/0.603 | 24.54/0.546 | 27.77/0.733 | 27.35/0.728 | 28.90/0.787 | 29.75/0.803 | **29.87/0.804** |
| Img8 | 10.41/0.079 | 14.15/0.186 | 16.63/0.491 | 15.66/0.443 | 15.69/0.190 | 17.43/0.459 | 17.58/0.465 | 18.66/0.519 | **18.77/0.539** | 18.46/0.531 |
| Aver. | 11.824 | 17.221 | 20.600 | 19.623 | 18.819 | 21.791 | 21.878 | 22.288 | **22.929** | 22.839 |

**Bold** fonts denote the best performance; underlined ones represent the second-best results in all tables.

From the comparison between Fig. 8 (d) and 8 (i) restored by NNM and QNNM, respectively, NNM contains quite blurry parts while LRQA-1 can restore more details, which directly verifies the advantage of the quaternion representation. D-N and F-N yield similar inpainting results with LRQA, however, the proposed LRQA-2 and LRQA-3 well preserve the light

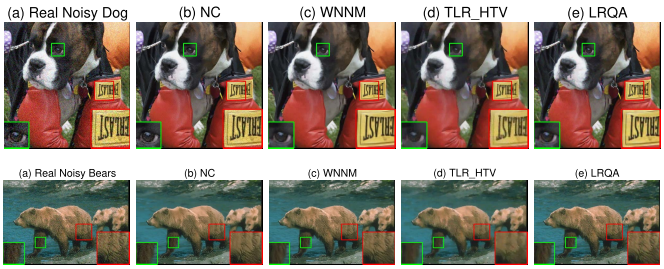


Fig. 6. Real color image denoising results on "Dog" and "Bears." (a) Real noisy image. The denoised image reconstructed by: (b) NC [60], (c) WNNM [53], (d) TLR-HTV [11], (e) LRQA. The figure is viewed better in zoomed PDF.

Fig. 7. The 8 color images for color image inpainting.

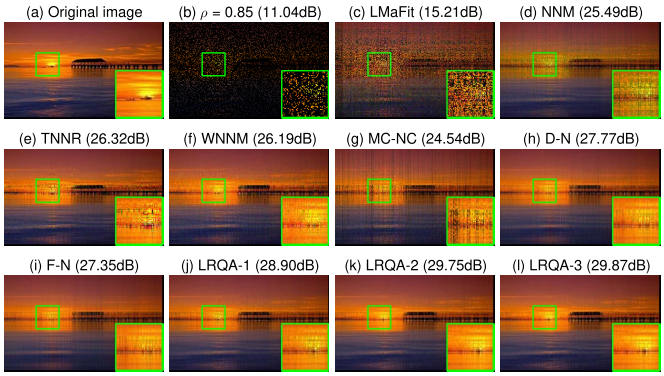


Fig. 8. Color image inpainting results on Img7 with 85% random missing pixels.

of the early morning sun (see the highlighted green rectangle in Fig. 8 (g) and (h)).

### D. Model Analysis

In this section, we aim to analyze the LRQA, including the parameter selection, empirical convergence, and runtime. Take the color image inpiainting with 85% random missing pixels as an example.

**Parameter selection:** There is only one free parameter $\gamma$ in the proposed Algorithms 2 and 3 based on the laplace function. For color image denoising, $\gamma$ is experimentally selected from the interval [0.1 1.15]. For color image inpainting, $\gamma$ is selected from [10 60]. Fig. 9 reports the PSNR and SSIM values of LRQA for color image denoising and inpainting with different $\gamma$. We can see that when $\gamma$ ranges from 0.5 to 1.15, LRMA performs better to remove noise, while when $\gamma$ ranges from 20 to 40, LRMA achieves promising performance for filling the missing pixels. The parameter analysis of other LRQA variants can be implemented in the similar way.

**Empirical convergence of LRQA:** We also plot the empirical convergence of LRQA for color image inpainting on all images as shown in Fig. 10, where the Y-axis denotes the relative error defined in Eq. (29). The convergence curves on

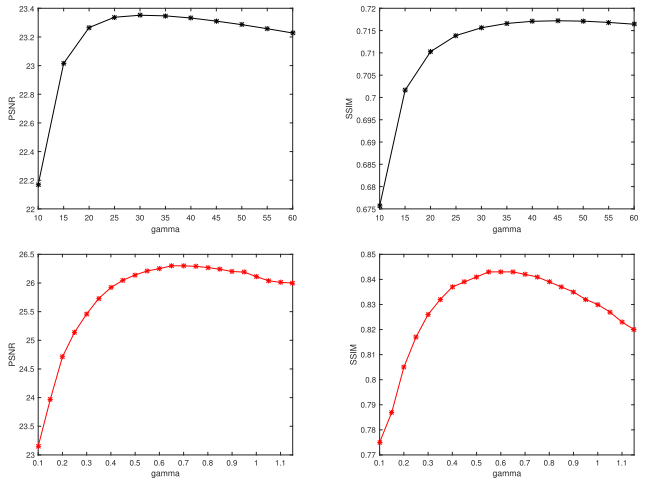


Fig. 9. PSNR and SSIM values of LRQA for color image inpainting with different $\gamma$ on Img6 (top) and PSNR and SSIM values of LRQA for color image denoising with different $\gamma$ on Img7 (bottom).

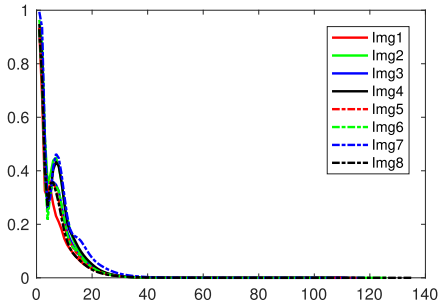


Fig. 10. Relative error versus iteration on all images.

TABLE V
AVERAGE RUNTIME (IN SECONDS) FOR COLOR IMAGE INPAINTING

| LMaFit [58] | NNM [16] | TNNR [4] | WNNM [1] | MC-NC [21] |
|---|---|---|---|---|
| 1.29 | 9.58 | 34.37 | 239.70 | 15.77 |
| D-N [13] | F-N [13] | LRQA-1 | LRQA-2 | LRQA-3 |
| 6.84 | 5.26 | 48.21 | 55.01 | 81.76 |

all images have a slight fluctuation around the 10-th iteration. But after 40 iterations, LRQA converges.

**Runtime:** To further investigate how LRQA efficiently processes color images, we report the average runtimes of all methods for color image inpainting on all images as shown in Table V. It can be seen that the matrix factorization-based methods, including LMaFit, D-N and F-N, are fastest. The reason is that they exploit the matrix factorization to depict the low-rank property and do not require computing SVD. WNNM is the slowest method since it not only needs to perform SVD, but also process three times on three color channels. The proposed LRQA achieves similar runtime with that of TNNR.

## VII. CONCLUSION

In this paper, we proposed a novel color image processing model, named low-rank quaternion approximation (LRQA),

which targets at recovering a low-rank quaternion matrix from its noisy observation. Different from the existing sparse representaion and low-rank matrix approximation (LRMA)-based methods, LRQA can process the three channels in a holistic way. Inspired by the excellent performance of the nonconvex functions for LRMA, we proposed a general model for LRQA, and then solve the optimization problem. Extensive experiments for color image denoising and inpainting tasks have demonstrated the effectiveness of the proposed LRQA. In the future, we would like to extend LRQA to other color image processing tasks, such as image deblurring, image super-resolution, image deraining and so on.

## APPENDIX A
## PROOF OF *Theorem 2*

Following [40], we need to prove that Eq. (7) has one unique optimal solution $\dot{\mathbf{X}}^{\#}$ and $\dot{\mathbf{X}}^{\#}$ equals to $\dot{\mathbf{X}}^{*}$ defined in Eq. (8). *Proof:* It is obvious that Eq. (7) has one unique optimal solution since these two terms of Eq. (7) are convex. The optimal solution is denoted as $\dot{\mathbf{X}}^{\#}$ and it must satisfy the following formula:

$$\dot{\mathbf{0}} \in \dot{\mathbf{X}}^{\#} - \dot{\mathbf{Y}} + \lambda \partial \|\dot{\mathbf{X}}^{\#}\|_{\circledast}, \tag{30}$$

where $\partial \|\dot{\mathbf{X}}^{\#}\|_{\circledast}$ denotes the subgradient of the quaternion nuclear norm at $\dot{\mathbf{X}}^{\#}$. For an arbitrary quaernion matrix $\dot{\mathbf{X}} \in \mathbb{H}^{m\times n}$, $\partial \|\dot{\mathbf{X}}\|_{\circledast}$ is defined as

$$\begin{aligned} \partial \|\dot{\mathbf{X}}\|_{\circledast} = \{&\dot{\mathbf{U}}\dot{\mathbf{V}}^{\triangleleft} + \dot{\mathbf{W}} : \dot{\mathbf{W}} \in \mathbb{H}^{m\times n}, \\ &\dot{\mathbf{U}}^{\triangleleft}\dot{\mathbf{W}} = \dot{\mathbf{0}}, \dot{\mathbf{W}}\dot{\mathbf{V}} = \dot{\mathbf{0}}, \|\dot{\mathbf{W}}\|_2 \le 1\}, \end{aligned} \tag{31}$$

where $\dot{\mathbf{X}} = \dot{\mathbf{U}}\Sigma\dot{\mathbf{V}}^{\triangleleft}$ and $\|\cdot\|_2$ is the spectral norm.

Next, we aim to prove that $\dot{\mathbf{X}}^{*}$ satisfies the equality in Eq. (30). The QSVD of $\dot{\mathbf{Y}}$ is

$$\dot{\mathbf{Y}} = \dot{\mathbf{U}}\Sigma\dot{\mathbf{V}}^{\triangleleft} = \dot{\mathbf{U}}_1\Sigma_1\dot{\mathbf{V}}_1^{\triangleleft} + \dot{\mathbf{U}}_2\Sigma_2\dot{\mathbf{V}}_2^{\triangleleft}, \tag{32}$$

where $\dot{\mathbf{U}} = [\dot{\mathbf{U}}_1, \dot{\mathbf{U}}_2]$, $\dot{\mathbf{V}} = [\dot{\mathbf{V}}_1, \dot{\mathbf{V}}_2]$, $\dot{\mathbf{U}}_1$, $\dot{\mathbf{V}}_1$ and $\Sigma_1$ are the singular values vectors and singular values corresponding to singular values larger than $\lambda$, while $\dot{\mathbf{U}}_2$, $\dot{\mathbf{V}}_2$ and $\Sigma_2$ correspond to singular values smaller than or equal to $\lambda$. Based on the above definitions and Eq. (8), $\dot{\mathbf{X}}^{*}$ can be reformulated as

$$\dot{\mathbf{X}}^{*} = \dot{\mathbf{U}}_1(\Sigma_1 - \lambda\dot{\mathbf{I}})\dot{\mathbf{V}}_1^{\triangleleft}. \tag{33}$$

then,

$$\dot{\mathbf{Y}} - \dot{\mathbf{X}}^{*} = \lambda(\dot{\mathbf{U}}_1\dot{\mathbf{V}}_1^{\triangleleft} + \dot{\mathbf{W}}), \quad \dot{\mathbf{W}} = \frac{1}{\lambda}\dot{\mathbf{U}}_2\Sigma_2\dot{\mathbf{V}}_2^{\triangleleft}. \tag{34}$$

Since $\dot{\mathbf{U}}$ and $\dot{\mathbf{V}}$ are unitary matrices, $\dot{\mathbf{U}}_1^{\triangleleft}\dot{\mathbf{U}}_2 = \dot{\mathbf{0}}$ and $\dot{\mathbf{V}}_2^{\triangleleft}\dot{\mathbf{V}}_1 = \dot{\mathbf{0}}$. Therefore, $\dot{\mathbf{U}}_1^{\triangleleft}\dot{\mathbf{W}} = \dot{\mathbf{0}}, \dot{\mathbf{W}}\dot{\mathbf{V}}_1 = \dot{\mathbf{0}}$. Since the singular values of $\Sigma_2$ are smller than or equal to $\lambda$, we have $\|\dot{\mathbf{W}}\|_2 \le 1$. From all above discussions, we can obtain that

$$\dot{\mathbf{0}} \in \dot{\mathbf{X}}^{*} - \dot{\mathbf{Y}} + \lambda \partial \|\dot{\mathbf{X}}^{*}\|_{\circledast}, \tag{35}$$

which means that $\dot{\mathbf{X}}^{*} = \dot{\mathbf{X}}^{\#}$. This completes the proof. ■

## APPENDIX B
## PROOF OF *Theorem 3*

*Proof:* $\dot{\mathbf{Y}} = \dot{\mathbf{U}}\Sigma_{\dot{\mathbf{Y}}}\dot{\mathbf{V}}^{\triangleleft}$ is the QSVD of $\dot{\mathbf{Y}} \in \mathbb{H}^{m\times n}$ and $\Sigma_{\dot{\mathbf{Y}}} = diag(\sigma(\dot{\mathbf{Y}})) = \dot{\mathbf{U}}^{\triangleleft}\dot{\mathbf{Y}}\dot{\mathbf{V}}$.

$$\frac{1}{2}\|\dot{\mathbf{Y}} - \dot{\mathbf{X}}\|_F^2 + \lambda\sum_i \phi(\sigma_i(\dot{\mathbf{X}})) \tag{36}$$

$$= \frac{1}{2}\|\Sigma_{\dot{\mathbf{Y}}} - \dot{\mathbf{U}}^{\triangleleft}\dot{\mathbf{X}}\dot{\mathbf{V}}\|_F^2 + \lambda\sum_i \phi(\sigma_i(\dot{\mathbf{X}})) \tag{37}$$

Eq. (37) is based on the fact that Frobenius norm is unitarily variant. Suppose $\dot{\mathbf{A}} = \dot{\mathbf{U}}^{\triangleleft}\dot{\mathbf{X}}\dot{\mathbf{V}}$ and $\dot{\mathbf{A}}$ and $\dot{\mathbf{X}}$ have the same singular values, thus Eq. (37) is equal to

$$= \frac{1}{2}\|\Sigma_{\dot{\mathbf{Y}}} - \dot{\mathbf{A}}\|_F^2 + \lambda\sum_i \phi(\sigma_i(\dot{\mathbf{A}})) \tag{38}$$

$$\ge \frac{1}{2}\|\Sigma_{\dot{\mathbf{Y}}} - \Sigma_{\dot{\mathbf{A}}}\|_F^2 + \lambda\sum_i \phi(\sigma_i(\dot{\mathbf{A}})) \tag{39}$$

$$= \frac{1}{2}\|\Sigma_{\dot{\mathbf{Y}}} - \Sigma_{\dot{\mathbf{X}}}\|_F^2 + \lambda\sum_i \phi(\sigma_i(\dot{\mathbf{X}})) \tag{40}$$

$$= \frac{1}{2}\|\sigma(\dot{\mathbf{Y}}) - \sigma(\dot{\mathbf{X}})\|_2^2 + \lambda\phi(\sigma(\dot{\mathbf{X}})) \tag{41}$$

$$\ge \frac{1}{2}\|\sigma(\dot{\mathbf{Y}}) - \sigma^{*}\|_2^2 + \lambda\phi(\sigma^{*}) \tag{42}$$

where $\sigma(\dot{\mathbf{Y}})$ and $\sigma(\dot{\mathbf{X}})$ are the singular values vectors of $\dot{\mathbf{Y}}$ and $\dot{\mathbf{X}}$, respectively. Using the Hoffman-Wielandt inequality, we can derive Eq. (39). Then, solving the original model in Eq. (9) is transformed by solving Eq. (42). Finally, the optimal solution $\dot{\mathbf{X}}^{*}$ is $\dot{\mathbf{X}}^{*} = \dot{\mathbf{U}}diag(\sigma^{*})\dot{\mathbf{V}}^{\triangleleft}$. ■

## REFERENCES


[1] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," *Int. J. Comput. Vis.*, vol. 121, no. 2, pp. 183–208, Jan. 2017.

[2] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, and L. Zhang, "Weighted schatten *p*-norm minimization for image denoising and background subtraction," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4842–4857, Oct. 2016.

[3] Y. Chen, Y. Guo, Y. Wang, D. Wang, C. Peng, and G. He, "Denoising of hyperspectral images using nonconvex low rank matrix approximation," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5366–5380, Sep. 2017.

[4] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2117–2130, Sep. 2013.

[5] H. Xue, S. Zhang, and D. Cai, "Depth image inpainting: Improving low rank matrix completion with low gradient regularization," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4311–4320, Sep. 2017.

[6] W. Ren, X. Cao, J. Pan, X. Guo, W. Zuo, and M.-H. Yang, "Image deblurring via enhanced low-rank prior," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3426–3437, Jul. 2016.

[7] S. Wang, Y. Wang, Y. Chen, P. Pan, Z. Sun, and G. He, "Robust PCA using matrix factorization for background/foreground separation," *IEEE Access*, vol. 6, pp. 18945–18953, 2018.

[8] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[9] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 1, pp. 225–253, 2014.

[10] X. Zhou, C. Yang, H. Zhao, and W. Yu, "Low-rank modeling and its applications in image analysis," *ACM Comput. Surv.*, vol. 47, no. 2, p. 36, 2015.

[11] Y. Chen, S. Wang, and Y. Zhou, "Tensor nuclear norm-based low-rank approximation with total variation regularization," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1364–1377, Dec. 2018.

[12] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.

[13] F. Shang, J. Cheng, Y. Liu, Z. Luo, and Z. Lin, "Bilinear factor matrix norm minimization for robust PCA: Algorithms and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 9, pp. 2066–2080, Sep. 2018.

[14] F. Xu, Y. Chen, C. Peng, Y. Wang, X. Liu, and G. He, "Denoising of hyperspectral image using low-rank matrix factorization," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1141–1145, Jul. 2017.

[15] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, 2002.

[16] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, p. 717, Dec. 2009.

[17] Z. Kang, C. Peng, and Q. Cheng, "Robust PCA via nonconvex rank approximation," in *Proc. IEEE Int. Conf. Data Mining*, Nov. 2015, pp. 211–220.

[18] L. E. Frank and J. H. Friedman, "A statistical view of some chemometrics regression tools," *Technometrics*, vol. 35, no. 2, pp. 109–135, 1993.

[19] Z. Kang, C. Peng, J. Cheng, and Q. Cheng, "LogDet rank minimization with application to subspace clustering," *Comput. Intell. Neurosci.*, vol. 2015, Jan. 2015, Art. no. 68.

[20] C. Lu, J. Tang, S. Yan, and Z. Lin, "Generalized nonconvex nonsmooth low-rank minimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 4130–4137.

[21] F. Nie, Z. Hu, and X. Li, "Matrix completion based on non-convex low-rank approximation," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2378–2388, May 2018.

[22] Y. Xu, L. Yu, H. Xu, H. Zhang, and T. Nguyen, "Vector sparse representation of color image using quaternion matrix analysis," *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1315–1329, Apr. 2015.

[23] J. Xu, L. Zhang, D. Zhang, and X. Feng, "Multi-channel weighted nuclear norm minimization for real color image denoising," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1096–1104.

[24] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2005, pp. 60–65.

[25] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognit.*, vol. 43, no. 4, pp. 1531–1549, 2010.

[26] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.

[27] S.-C. Pei and C.-M. Cheng, "Color image processing by using binary quaternion-moment-preserving thresholding technique," *IEEE Trans. Image Process.*, vol. 8, no. 5, pp. 614–628, May 1999.

[28] F. Zhang, "Quaternions and matrices of quaternions," *Linear Algebra Appl.*, vol. 251, pp. 21–57, Jan. 1997.

[29] S. J. Sangwine and N. L. Bihan, "Quaternion singular value decomposition based on bidiagonalization to a real or complex matrix using quaternion Householder transformations," *Appl. Math. Comput.*, vol. 182, no. 1, pp. 727–738, 2006.

[30] X. Xiao and Y. Zhou, "Two-dimensional quaternion PCA and sparse PCA," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 2028–2042, Jul. 2018.

[31] R. Lan and Y. Zhou, "Quaternion-michelson descriptor for color image classification," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5281–5292, Nov. 2016.

[32] N. Le Bihan and S. J. Sangwine, "Quaternion principal component analysis of color images," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, Sep. 2003, p. I-809.

[33] C. Zou, K. Kou, and Y. Wang, "Quaternion collaborative and sparse representation with application to color face recognition," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3287–3302, Jul. 2016.

[34] R. Zeng, J. Wu, Z. Shao, Y. Chen, B. Chen, L. Senhadji, and H. Shu, "Color image classification via quaternion principal component analysis network," *Neurocomputing*, vol. 216, pp. 416–428, Dec. 2016.

[35] X. Zhu, Y. Xu, H. Xu, and C. Chen, "Quaternion convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 631–647.

[36] R. Lan, Y. Zhou, and Y. Y. Tang, "Quaternionic weber local descriptor of color images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 2, pp. 261–274, Feb. 2017.

[37] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[38] T. Parcollet, M. Morchid, and G. Linarès, "Quaternion convolutional neural networks for heterogeneous image processing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2019, pp. 8514–8518.

[39] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, p. 11, May 2011.

[40] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.

[41] Y. Chen, X. Xiao, and Y. Zhou, "Multi-view clustering via simultaneously learning graph regularized low-rank tensor representation and affinity matrix," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2019, pp. 1348–1353.

[42] A. E. Waters, A. C. Sankaranarayanan, and R. Baraniuk, "SpaRCS: Recovering low-rank and sparse matrices from compressive measurements," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1089–1097.

[43] Q. Sun, S. Xiang, and J. Ye, "Robust principal component analysis via capped norms," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 311–319.

[44] T.-H. Oh, Y.-W. Tai, J.-C. Bazin, H. Kim, and I. S. Kweon, "Partial sum minimization of singular values in robust PCA: Algorithm and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 744–758, Apr. 2016.

[45] M. Fazel, H. Hindi, and S. P. Boyd, "Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices," in *Proc. IEEE Amer. Control Conf.*, vol. 3, Jun. 2003, pp. 2156–2162.

[46] Q. Feng *et al.*, "Superimposed sparse parameter classifiers for face recognition," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 378–390, Feb. 2017.

[47] W. R. Hamilton, *Elements of Quaternions*. Harlow, U.K.: Longmans, 1866.

[48] R. D. Schafer, "On the algebras formed by the Cayley–Dickson process," *Amer. J. Math.*, vol. 76, no. 2, pp. 435–446, 1954.

[49] P. R. Girard, *Quaternions, Clifford Algebras and Relativistic Physics*. Springer, 2007.

[50] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, 2011.

[51] P. D. Tao and L. T. H. An, "Convex analysis approach to DC programming: Theory, algorithms and applications," *Acta Math. Vietnamica*, vol. 22, no. 1, pp. 289–355, 1997.

[52] M. Rizkinia, T. Baba, K. Shirai, and M. Okuda, "Local spectral component decomposition for multi-channel image denoising," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3208–3218, Jul. 2016.

[53] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.

[54] W. Dong, G. Shi, and X. Li, "Nonlocal image restoration with bilateral variance estimation: A low-rank approach," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 700–711, Feb. 2013.

[55] Y. Chen, Y. Wang, M. Li, and G. He, "Augmented Lagrangian alternating direction method for low-rank minimization via non-convex approximation," *Signal, Image Video Process.*, vol. 11, no. 7, pp. 1271–1278, 2017.

[56] L. Li, X. Wang, and G. Wang, "Alternating direction method of multipliers for separable convex optimization of real functions in complex variables," *Math. Problems Eng.*, vol. 2015, Nov. 2015, Art. no. 104531.

[57] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[58] Y. Shen, Z. Wen, and Y. Zhang, "Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization," *Optim. Methods Softw.*, vol. 29, no. 2, pp. 239–263, 2014.

[59] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[60] M. Lebrun, M. Colom, and J.-M. Morel, "Multiscale image blind denoising," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3149–3161, Oct. 2015.

**Yongyong Chen** received the B.S. and M.S. degrees from the College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao, China. He is currently pursuing the Ph.D. degree with the Department of Computer and Information Science, University of Macau, Macau, China. He visited the National Key Lab for Novel Software Technology, Nanjing University, as an Exchange Student in 2017. His research interests include (non-convex) low-rank and sparse matrix/tensor decomposition models, with applications to image processing, data mining, and computer vision.

**Xiaolin Xiao** received the B.E. degree from Wuhan University, China, in 2013, and the Ph.D. degree from the University of Macau, Macau, China, in 2019. She is currently a Postdoctoral Fellow with the School of Computer Science and Engineering, South China University of Technology, China. Her research interests include superpixel segmentation, saliency detection, and color image processing and understanding.

**Yicong Zhou** (M'07–SM'14) received the B.S. degree from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees from Tufts University, Massachusetts, USA, all in electrical engineering.

He is currently an Associate Professor and the Director of the Vision and Image Processing Laboratory, Department of Computer and Information Science, University of Macau. His research interests include image processing and understanding, computer vision, machine learning, and multimedia security.

Dr. Zhou is a Senior Member of the International Society for Optical Engineering (SPIE). He was a recipient of the Third Price of Macau Natural Science Award in 2014. He is a Co-Chair of Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society. He serves as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and four other journals.