Pattern Recognition 128 (2022) 108665

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Quaternion-based weighted nuclear norm minimization for color image restoration

Chaoyan Huang[a], Zhi Li[b], Yubing Liu[c], Tingting Wu[a,*], Tieyong Zeng[d]

[a] School of Science, Nanjing University of Posts and Telecommunications, Nanjing, 210023, China
[b] The Department of Computer Science and Technology, Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai 200241, China
[c] Bell honor school, Nanjing University of Posts and Telecommunications, Nanjing, 210023, China
[d] Department of Mathematics, The Chinese University of Hong Kong, Shatin, NT, Hong Kong

ARTICLE INFO

Article history:
Received 13 January 2021
Revised 20 January 2022
Accepted 21 March 2022
Available online 23 March 2022

Keywords:
Quaternion representation
Color image restoration
Weighted nuclear norm
Variational method
Low-rank matrix analysis

ABSTRACT

Color image restoration is one of the basic tasks in pattern recognition. Unlike grayscale image, each color image has three channels in the RGB color space. Due to the inner-relationship within the three channels, color image restoration is usually much more difficult than its grayscale counterpart. Indeed, new problems such as color artifacts could emerge when the grayscale image processing methods are extended to color images directly. Note that one of the most effective gray image restoration methods is the weighted nuclear norm minimization (WNNM) approach. However, when applied to color images, the results of WNNM are usually not as promising as that of grayscale images. In order to solve this problem, in this paper, we propose to restore color images with the quaternion-based WNNM method (QWNNM) since the structure of color channels can be well preserved with quaternion representation. The proposed model can be solved efficiently by the alternating direction method of multipliers (ADMM). The theoretical analysis of the optimal solution is also presented. Numerical experiments are carefully conducted with different kinds of degradation to illustrate the superior performance of our proposed QWNNM over the state-of-the-art methods, including a celebrated deep learning approach, in both visual quality and numerical results.

1. Introduction

Color images corrupted by blur and noise are unavoidable in the process of image storage and transmission. Given the rapidly growing role of color images in daily lives, color image restoration has become a popular research field. In the literature, remarkable efforts have been put to recover color images from their corresponding degraded ones, such as the variation-based methods [1], the dictionary-based methods [2], and the convolutional neural network-based methods [3,4], etc. Most of the above methods are to handle the color image in the real number domain [5]. It is well known that color images consist of three color channels (red, green, and blue in the RGB color space) with inner-relationships among them [6]. If the three color channels in the real number space are treated as three stand-alone matrices [7], the negligence of the inner-relationship among color channels is inevitable. Recently, due to its capability of well preserving the structure of channels, the quaternion representation [8] has been widely used in color image processing. For instance, Subakan and Vemuri [9] presented color image smoothing and segmentation with the quaternion framework. Li et al. [10,11] handled the color image denoising task with the quaternion-based non-local total variation and non-local means. Chen et al. [12] and Yu et al. [13] denoised color images with quaternion-based low-rank regularizer. Shi et al. [14] proposed a color histopathological image classification method based on quaternion Grassmann average network. Quaternion neural networks were also applied to image denoising [15] and classification [16]. However, color image deblurring in the quaternion domain remains a challenging problem. The difficulty of the task lies in how the blurring operator is dealt with. In the quaternion domain, the special multiplication rule makes the deblurring task even harder. Restoring color images in the quaternion domain directly is still a big challenge and an open question. As far as we know, Jia et al. [17] proposed a color image restoration model based on quaternion by transforming the color image into the HSV color space instead of handling the quaternion representation directly. In this paper, we propose a new quaternion-based

* Corresponding author.
E-mail address: wutt@njupt.edu.cn (T. Wu).

https://doi.org/10.1016/j.patcog.2022.108665
0031-3203/

color image restoration method. To solve the proposed model directly in the quaternion domain, we design a special quaternion blurring matrix.

Restoring a color image from the corresponding corrupted one is usually complicated [18,19]. Due to the fact that the high-dimensional matrix inherently has a low-rank structure [20], restricting the underlying image matrix to be low rank is often a better approach [21]. The low-rank image restoration methods usually outperform methods based on nonlocal and block-matching and 3D filtering (BM3D) [22]. The nuclear norm minimization (NNM) is a common convex surrogate to the rank minimization problem. The NNM method treats every singular value equally, which means that it posits all the singular values contain the same information, regardless of their sizes. However, many works [23,24] have pointed out that the larger singular values carry more data, which explains why the NNM yields only sub-optimal results. Gu et al. [25] proposed a method based on weighted nuclear norm minimization (WNNM) which assigned a weight to each singular value for grayscale image denoising. In 2016, Gu et al. [23] extended the application of the WNNM method, including grayscale image denoising, background subtraction, color image inpainting, and color image denoising. In the results of the image denoising schemes, an interesting phenomenon is observed, namely that some color spots remain in the denoising results of color images, while the grayscale images have almost perfect results. Interested readers may look up the reference [23] (Fig. 5) for details.

The underlying difference between color and grayscale images is the color channels. If color images are denoted as the combination of grayscale images, the inner-relationships between different channels will be totally ignored [26]. Moreover, it is not reasonable to handle the color image with a low-rank approximation method by representing the color image as the linear combination of grayscale images. The three grayscale images that correspond to each channel have different singular value matrices. When we only consider these singular values, the correlation between the three matrices is not taken into account. Whereas representing the color image as a pure quaternion matrix, all we need is to deal with a quaternion singular value matrix. The relationship between the color channels can also be well utilized, hence the degraded image can be better restored [27]. Given the similarity between the three color channels in color images and the three imaginary parts in pure quaternion matrices, we represent color images with pure quaternion matrices [28] in this paper. On the other hand, quaternion matrices have a special multiplication rule, which can be exploited to better present the internal structure of color images [29]. Many researchers have presented theoretical analyses of the quaternions. Quaternion Fourier transform was studied in [30]. Zhang gave some brief proofs of theories on quaternions and quaternion matrices [28]. The theory of quaternion matrix derivatives was studied in [31]. In 2017, Jia et al. [32] presented the theory of the quaternion-based principal component analysis. Later, Chen et al. [33] incorporated the singular value decomposition into the quaternion domain. Qi and Zhang [34] extended the theory of complex number domain into quaternion domain. These rich mathematical tools grant us theoretical guarantees when quaternion representation of color images is used.

To ensure the color image restoration tasks can be well implemented, we represent the color image with a pure quaternion matrix. A preliminary result of the proposed model is displayed in Fig. 1. To better demonstrate the visual quality and numerical results of the proposed model, we compare our model with the convolutional neural network-based method IRCNN [35]. From the restoration results, we can see that the quaternion representation is better than the real-valued ones, even if it is a CNN-based method. Different from existing color image restoration methods, our main contributions are as follows.

- We develop the classical WNNM method to the quaternion domain for color image restoration. We encode the color image as a pure quaternion matrix, such that more interconnected information among the RGB channels can be preserved.
- To better recover the color image in the quaternion domain, we extend the 2-dimension blurring matrix to the quaternion domain and design a quaternion blurring operator.
- We solve the proposed model efficiently by the ADMM method and the theoretical analysis of the uniqueness of the solution is also presented.
- Experiments show that the proposed method outperforms some state-of-the-art methods, including the CNN-based method.

The outline of this paper is as follows. Section 2 reviews the basic concepts of quaternion algebra and the classical WNNM method. Section 3 presents the proposed model and theoretical analysis. In Section 4, numerical experiments and results are presented and reported. Finally, we conclude our paper in Section 5.

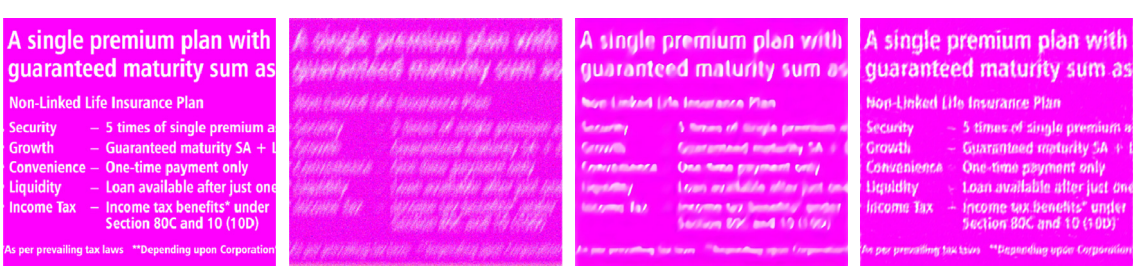
2. Related concepts

2.1. Quaternion

Let $\mathbb{R}$ be the real number space with real numbers denoted by a, and $\mathbb{H}$ be the quaternion space with quaternions denoted by $\dot{a}$. The quaternion was first introduced by Hamilton [8]. Suppose $\dot{a}$ is a quaternion, then it can be written as

$$\dot{a} = a_0 + a_1 i + a_2 j + a_3 k, \tag{1}$$

where a_0, a_1, a_2, $a_3 \in \mathbb{R}$ and i, j, k are the fundamental quaternion units which satisfy the quaternion rules $i^2 = j^2 = k^2 = ijk = -1$. More specifically, the $ij = k$, $jk = i$, $ki = j$, $ji = -k$, $kj = -i$, $ik = -j$. Form $ij = k$, we know that $ijk = kk = k^2 = -1$ [8]. Let $\dot{a} = a_0 + a_1 i + a_2 j + a_3 k \in \mathbb{H}$, $\dot{b} = b_0 + b_1 i + b_2 j + b_3 k \in \mathbb{H}$, and



(a) Img26 (b) Degraded (14.59/0.8183) (c) IRCNN (17.85/0.9196) (h) Our (20.34/0.9562)

Fig. 1. Color image restoration on Img26 with visual quality and numerical results (PSNR/SSIM). (a) Original image, (b) degraded image with motion kernel (20, 60) and Gaussian noise level $\sigma = 25$, restored image reconstructed by: (c) IRCNN [35], (h) Our QWNNM.

$\lambda \in \mathbb{R}$, then we have

$$\dot{a} + \dot{b} = (a_0 + b_0) + (a_1 + b_1)i + (a_2 + b_2)j + (a_3 + b_3)k, \tag{2}$$

$$\lambda\dot{a} = (\lambda a_0) + (\lambda a_1)i + (\lambda a_2)j + (\lambda a_3)k,$$

and

$$\dot{a}\dot{b} = (a_0b_0 - a_1b_1 - a_2b_2 - a_3b_3) + (a_0b_1 + a_1b_0 + a_2b_3 - a_3b_2)i \tag{3}$$
$$+ (a_0b_2 - a_1b_3 + a_2b_0 + a_3b_1)j + (a_0b_3 + a_1b_2 - a_2b_1 + a_3b_0)k.$$

The conjugate and modulus of $\dot{a}$ are defined by

$$\dot{a}^* = a_0 - a_1 i - a_2 j - a_3 k, \tag{4}$$

$$|\dot{a}| = \sqrt{a_0^2 + a_1^2 + a_2^2 + a_3^2}.$$

In particular, a pure quaternion $\dot{c}$ is defined as

$$\dot{c} = c_1 i + c_2 j + c_3 k, \tag{5}$$

which means the real part of a pure quaternion is equal to zero. A quaternion matrix can be written as $\dot{D} = D_0 + D_1 i + D_2 j + D_3 k \in \mathbb{H}^{m\times n}$. According to Eq. (1), we see that a quaternion has three imaginary parts. Since color images are consist of three color channels, we represent a color image as a pure quaternion matrix, with each pixel of an image considered as a pure quaternion (5). Suppose $\dot{u}$ is a color image with quaternion matrix representation with three components (red, green, and blue). Mathematically, we indicate $\dot{u}$ as

$$\dot{u} = u_r i + u_g j + u_b k, \tag{6}$$

where u_r, u_g, and u_b are the RGB channels of $\dot{u}$ respectively. Every pixel of $\dot{u}$ is a pure quaternion.

The identity quaternion matrix $\dot{I}$ is the same as the classical identity matrix. For example, if $\dot{I} \in \mathbb{H}^{2\times 2}$, then

$$\dot{I} = \begin{pmatrix} 1 + 0i + 0j + 0k & 0 + 0i + 0j + 0k \\ 0 + 0i + 0j + 0k & 1 + 0i + 0j + 0k \end{pmatrix}. \tag{7}$$

A square quaternion matrix is unitary if $\dot{D}^*\dot{D} = \dot{D}\dot{D}^* = \dot{I}$. The norms of quaternion matrices and vectors are defined as follows.

Definition 2.1. The ℓ_2-norm of quaternion vector $\dot{a} = \alpha_0 + \alpha_1 i + \alpha_2 j + \alpha_3 k \in \mathbb{H}^n$ is $\|\dot{a}\|_2 := \sqrt{\sum_i |a_i|^2}$; the ℓ_2-norm of quaternion matrix is $\|\dot{D}\|_2 := \max(\sigma(\dot{D}))$, where $\sigma(\dot{D})$ is the set of singular values of $\dot{D}$, and the Frobenius norm is $\|\dot{D}\|_F := \sqrt{\sum_{i,j} |D_{i,j}|^2}$.

Definition 2.2. The norm $\|\dot{D}\|_2 = \sqrt{\mathrm{tr}(\dot{D}^*\dot{D})}$, where $\mathrm{tr}(\cdot)$ is the trace of matrix. $\mathrm{tr}(D) = \sum_{i=1}^{\min(m,n)} \delta_i$, δ_i are the singular values of $\dot{D}$, $i = 1,2,\cdots, \min(m,n)$.

The singular value decomposition (SVD) of a quaternion matrix was proposed in [28] firstly.

Theorem 2.1. ***(Quaternion Singular Value Decomposition (QSVD))*** *Let $\dot{D} \in \mathbb{H}^{m\times n}$, then there exist two unitary quaternion matrices $\dot{U} \in \mathbb{H}^{m\times m}$ and $\dot{V} \in \mathbb{H}^{n\times n}$ such that $\dot{U}\dot{D}\dot{V}^* = \Sigma$, where $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s)$, $\sigma_i \geq 0$ are the singular values of $\dot{D}$ and $s = \min(m,n)$.*

2.2. WNNM regularization

The weighted nuclear norm minimization (WNNM) method was first proposed by Gu et al. [25] for grayscale image denoising. Their model can be written as

$$\min_u \frac{\lambda}{2}\|u - z\|_2^2 + \|u\|_{w,*}, \tag{8}$$

where λ is a positive parameter, u is the ideal image, z is the observed image, and $\|\cdot\|_{w,*}$ denotes the weighted nuclear norm. The model (8) has good performance in grayscale image denoising. Xie et al. gave the optimal solution of (8) [36]. Later, Gu et al. showed some applications of the WNNM method [23]. Due to the good performance and the existence of the optimal solution, the WNNM regularization was widely used in image processing. Ma et al. [37] combined the WNNM and total variation regularizer for grayscale image deblurring. Yair and Michaeli [38] applied the multi-scale strategy with WNNM for image restoration. [39] developed a model with a new data fidelity term to tackle mixed noise images.

All the aforementioned WNNM-based methods handle grayscale or color image restoration in a monochromatic way. In other words, they treat the three channels of the color image as three independent images and overlook the relationship between them. In this paper, we represent color images with pure quaternion matrices and extend the WNNM to the quaternion domain, which can better preserve the color structure and generate better color image restoration results.

3. Proposed model

In order to better preserve the inner-relationship of color channels, we represent the color image as a pure quaternion matrix. More specifically, suppose $\dot{u} \in \mathbb{H}^{m\times n}$ is a color image, the corresponding pure quaternion representation is as

$$\dot{u} = \begin{pmatrix} \dot{u}_{11}, \dot{u}_{12}, \cdots, \dot{u}_{1n} \\ \dot{u}_{21}, \dot{u}_{22}, \cdots, \dot{u}_{2n} \\ \vdots \\ \dot{u}_{m1}, \dot{u}_{m2}, \cdots, \dot{u}_{mn} \end{pmatrix}, \tag{9}$$

where pure quaternion $\dot{u}_{ij} = \dot{u}_{ij}^r i + \dot{u}_{ij}^g j + \dot{u}_{ij}^b k$ denotes a pixel of $\dot{u}$, $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, n$, here $\dot{u}_{ij}^r$, $\dot{u}_{ij}^g$, and $\dot{u}_{ij}^b$ are the red, green, and blue components, respectively. Suppose that $\dot{f} \in \mathbb{H}^{m\times n}$ is the observed image with quaternion representation, and $\dot{u} \in \mathbb{H}^{m\times n}$ is the ideal image with quaternion representation, our method can be written as follows

$$\min_{\dot{u}} \frac{\lambda}{2}\|\dot{A}\dot{u} - \dot{f}\|_2^2 + \|\dot{u}\|_{w,*}, \tag{10}$$

where $\dot{A}$ is a linear operator with quaternion representation and $\|\cdot\|_{w,*}$ denotes the weighted nuclear norm. The weights w_i for the ith singular of $\dot{u}$ follows the setting of WNNM [23] as

$$w_i = \frac{c}{\sigma_i + \epsilon}, \tag{11}$$

where c is a compromising constant, σ_i is the ith singular of $\dot{u}$, and ϵ is a small positive number.

Due to the blur operation and the weighted nuclear norm, it is not easy to directly solve model (10). Thanks to the alternating direction method of multipliers (ADMM) solver, our model can be handled and get the exact solution, so that the restoration results become more robust. By introducing the quaternion auxiliary variable $\dot{g} \in \mathbb{H}^{m\times n}$ and quaternion multiplier $\dot{\eta} \in \mathbb{H}^{m\times n}$ and letting $\dot{u} = \dot{g}$, the augmented Lagrangian of (10) is formulated by

$$\mathcal{L}(\dot{u}, \dot{g}; \dot{\eta}) = \min_{\dot{u},\dot{g},\dot{\eta}} \frac{\lambda}{2}\|\dot{A}\dot{u} - \dot{f}\|_2^2 + \|\dot{g}\|_{w,*} + \frac{\beta}{2}\|\dot{u} - \dot{g}\|_2^2 + \langle \dot{\eta}, \dot{u} - \dot{g}\rangle, \tag{12}$$

where $\beta > 0$ is a penalty parameter. The ADMM iteration for solving (12) goes as follows

$$\begin{cases} \dot{u}^{k+1} = \arg\min_{\dot{u}} \mathcal{L}(\dot{u}, \dot{g}^k; \dot{\eta}^k), \\ \dot{g}^{k+1} = \arg\min_{\dot{g}} \mathcal{L}(\dot{u}^{k+1}, \dot{g}; \dot{\eta}^k), \\ \dot{\eta}^{k+1} = \dot{\eta}^k + (\dot{u}^{k+1} - \dot{g}^{k+1}). \end{cases} \tag{13}$$

Table 1
Derivatives of Function of the Type $h(\dot{u})$ in [31].

$h(\dot{u})$	$\mathcal{D}_{\dot{u}}h$	Note
$\dot{u}$	1	$\dot{u} \in \mathbb{H}$
$\dot{\mu}\dot{u}$	$\dot{\mu}$	$\forall \dot{\mu} \in \mathbb{H}$
$\dot{u}\dot{v}$	$\Re(\dot{v})$	$\forall \dot{v} \in \mathbb{H}$, $\Re(\dot{v})$ denotes the real part of $\dot{v}$
$\dot{\mu}\dot{u}\dot{v} + \dot{\tau}$	$\dot{\mu}\Re(\dot{v})$	$\forall \dot{\mu}, \dot{v}, \dot{\tau} \in \mathbb{H}$
$\dot{u}^*$	$-\frac{1}{2}$	$\dot{u}^*$ denotes the conjugation of $\dot{u}$
$\dot{\mu}\dot{u}^*$	$-\frac{1}{2}\dot{\mu}$	$\forall \dot{\mu} \in \mathbb{H}$
$\dot{u}^*\dot{v}$	$-\frac{1}{2}\dot{v}^*$	$\forall \dot{v} \in \mathbb{H}$
$\dot{\mu}\dot{u}^*\dot{v} + \dot{\tau}$	$-\frac{1}{2}\dot{\mu}\dot{v}^*$	$\forall \dot{\mu}, \dot{v}, \dot{\tau} \in \mathbb{H}$
$\dot{u}^{-1}$	$-\dot{u}^{-1}\Re(\dot{u}^{-1})$	$\dot{u}^{-1}$ denotes the reciprocal of $\dot{u}$
$(\dot{u}^*)^{-1}$	$\frac{1}{2\lvert\dot{u}\rvert^2}$	–
$(\dot{\mu}\dot{u}\dot{v} + \dot{\tau})^2$	$\dot{g}\dot{\mu}\Re(\dot{v}) + \dot{\mu}\Re(\dot{v}\dot{g})$	$\dot{g} = \dot{\mu}\dot{u}\dot{v} + \dot{\tau}$
$(\dot{\mu}\dot{u}^*\dot{v} + \dot{\tau})^2$	$-\frac{1}{2}\dot{g}\dot{\mu}\dot{v}^* - \frac{1}{2}\dot{\mu}(\dot{v}\dot{g})^*$	$\dot{g} = \dot{\mu}\dot{u}^*\dot{v} + \dot{\tau}$
$\lvert\dot{\mu}\dot{u}\dot{v} + \dot{\tau}\rvert$	$\frac{\dot{g}^*}{2\lvert\dot{g}\rvert}\dot{\mu}\Re(\dot{v}) - \frac{1}{4\lvert\dot{g}\rvert}\dot{v}^*(\dot{\mu}^*\dot{g})^*$	$\dot{g} = \dot{\mu}\dot{u}\dot{v} + \dot{\tau}$
$\lvert\dot{\mu}\dot{u}^*\dot{v} + \dot{\tau}\rvert$	$\frac{\dot{g}}{2\lvert\dot{g}\rvert}\dot{v}^*\Re(\dot{\mu}^*) - \frac{1}{4\lvert\dot{g}\rvert}\dot{\mu}(\dot{v}\dot{g}^*)^*$	$\dot{g} = \dot{\mu}\dot{u}^*\dot{v} + \dot{\tau}$
$\lvert\dot{\mu}\dot{u}\dot{v} + \dot{\tau}\rvert^2$	$\dot{g}^*\dot{\mu}\Re(\dot{v}) - \frac{1}{2}\dot{v}^*(\dot{\mu}^*\dot{g})^*$	$\dot{g} = \dot{\mu}\dot{u}\dot{v} + \dot{\tau}$
$\lvert\dot{\mu}\dot{u}^*\dot{v} + \dot{\tau}\rvert^2$	$\dot{g}\dot{v}^*\Re(\dot{\mu}^*) - \frac{1}{2}\dot{\mu}(\dot{v}\dot{g}^*)^*$	$\dot{g} = \dot{\mu}\dot{u}^*\dot{v} + \dot{\tau}$

Next, we elaborate on how to solve these subproblems respectively. The $\dot{u}$-subproblem is written as

$$\dot{u}^{k+1} = \arg\min_{\dot{u}} \frac{\lambda}{2}\|\dot{A}\dot{u} - \dot{f}\|_2^2 + \frac{\beta}{2}\|\dot{u} - \dot{g}^k\|_2^2 + \langle \dot{\eta}^k, \dot{u} - \dot{g}^k\rangle. \tag{14}$$

According to the theory of quaternion matrix derivatives [31] (here we list some required rules of quaternion matrix derivatives in Table 1), then the optimality condition of (14) is given by

$$\frac{\lambda}{2}(\dot{A}\dot{u} - \dot{f})^*\dot{A} - \frac{\lambda}{4}(\dot{A}^*(\dot{A}\dot{u} - \dot{f}))^* + \frac{\beta}{4}\left(\dot{u} - \dot{g}^k + \frac{\dot{\eta}^k}{\beta}\right)^* = 0, \tag{15}$$

where $(\cdot)^*$ denotes the conjugation operator. We can obtain the following linear system

$$(\lambda\dot{A}^*\dot{A} + \beta)\dot{u} = \lambda\dot{A}^*\dot{f} + \beta\dot{g}^k - \dot{\eta}^k. \tag{16}$$

Under the quaternion periodic boundary condition for $\dot{u}$, we know that $\dot{A}^*\dot{A}$, and $\dot{A}^*\dot{f}$ are block circulant, so the quaternion fast Fourier transform [30] can be used to find the solution of (16)

$$\dot{u}^{k+1} = \mathcal{F}^{-1}\left(\frac{\lambda\mathcal{F}(\dot{A})^* \circ \mathcal{F}(\dot{f}) + \beta\mathcal{F}(\dot{g}) - \mathcal{F}(\dot{\eta})}{\lambda\mathcal{F}(\dot{A})^* \circ \mathcal{F}(\dot{A}) + \beta}\right), \tag{17}$$

where $\mathcal{F}$ denotes two-dimensional discrete quaternion Fourier transform, $\mathcal{F}^{-1}$ represents two-dimensional discrete inverse quaternion Fourier transform, $\circ$ means component-wise multiplication, and the division is component-wise as well.

The $\dot{g}$-subproblem is equivalent to

$$\dot{g}^{k+1} = \arg\min_{\dot{g}} \|\dot{g}\|_{w,*} + \frac{\beta}{2}\|\dot{g} - \left(\dot{u}^{k+1} + \frac{\dot{\eta}^k}{\beta}\right)\|_2^2. \tag{18}$$

Before solving the $\dot{g}$-subproblem, we first give the following theorems.

Theorem 3.1. *For any $\dot{B} \in \mathbb{H}^{m\times n}$ (without loss of generality we assume that $m \geq n$) and a diagonal non-negative matrix $E \in \mathbb{R}^{m\times m}$, the diagonal elements e_i follows $e_1 \geq e_2 \geq \cdots e_n \geq 0$. Let $\dot{B} = \dot{X}\Phi\dot{Y}^*$, we have*

$$\sum_{i=1}^{n} e_i t_i = \max_{\dot{U}^*\dot{U}=\dot{I}, \dot{V}^*\dot{V}=\dot{I}} \mathrm{tr}(E\dot{U}^*\dot{B}\dot{V}), \tag{19}$$

where t_i and e_i are the ith singular values of $\dot{B}$ and E, respectively. When $\dot{U} = \dot{X}$ and $\dot{V} = \dot{Y}$, $\mathrm{tr}(E\dot{U}^*\dot{B}\dot{V})$ reaches its maximum value.

Proof. Denote λ_i, $i = 1, 2, \cdots, n$, are the singular values of $E\dot{U}^*\dot{B}\dot{V}$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$.

$$\mathrm{tr}(E\dot{U}^*\dot{B}\dot{V}) = \sum_{i=1}^{n}\lambda_i \leq \sum_{i=1}^{n} e_i t_i, \tag{20}$$

for $\dot{U} = \dot{X}$ and $\dot{V} = \dot{Y}$, $\mathrm{tr}(E\dot{U}^*\dot{B}\dot{V})$ reaches its maximum value, i.e., Eq. (19) holds. □

Theorem 3.2. *For any $\dot{p} \in \mathbb{H}^{m\times n}$, the QSVD of $\dot{p}$ is $\dot{p} = \dot{U}S\dot{V}^*$, where*

$$S = \begin{pmatrix} \mathrm{diag}(s_1, s_2, \ldots, s_n) \\ 0 \end{pmatrix}. \tag{21}$$

According to (18), $\dot{p} = \dot{u} + \frac{\dot{\eta}}{\beta}$. The solution of $\dot{g}$-subproblem is $\dot{g} = \dot{U}K\dot{V}^*$, where

$$K = \begin{pmatrix} \mathrm{diag}(k_1, k_2, \ldots, k_n) \\ 0 \end{pmatrix}, \tag{22}$$

and $(k_1, k_2, \ldots, k_n)$ is the solution of the following convex optimization problem

$$\min_{k_1,\ldots,k_n} \sum_{i=1}^{n}(k_i - s_i)^2 + w_i k_i, \text{ s.t. } k_1 \geq k_2 \geq \cdots \geq k_n \geq 0. \tag{23}$$

Proof. For any $\dot{g} \in \mathbb{H}^{m\times n}$, according to Theorem 2.1, we have $\dot{g} = \dot{U}_1 K \dot{V}_1^*$ as the singular value decomposition of $\dot{g}$ and

$$K = \begin{pmatrix} \mathrm{diag}(k_1, k_2, \ldots, k_n) \\ 0 \end{pmatrix}, \tag{24}$$

with $k_1 \geq k_2 \geq \cdots \geq k_n \geq 0$. From $\dot{p} = \dot{u} + \frac{\dot{\eta}}{\beta}$, we know that

$$
\begin{aligned}
&\min_{\dot{g}} \|\dot{g}\|_{w,*} + \frac{\beta}{2}\|\dot{g} - (\dot{u} + \frac{\dot{\eta}}{\beta})\|_2^2 \\
\Leftrightarrow &\min_{\dot{g}} \|\dot{g}\|_{w,*} + \frac{\beta}{2}\|\dot{g} - \dot{p}\|_2^2 \\
\Leftrightarrow &\min_{\substack{\dot{U}_1,K,\dot{V}_1 \\ k_1\geq k_2\geq\cdots\geq k_n\geq 0}} \|\dot{U}_1 K \dot{V}_1^*\|_{w,*} + \|\dot{p} - \dot{U}_1 K \dot{V}_1^*\|_2^2 \\
\Leftrightarrow &\min_{\substack{\dot{U}_1,K,\dot{V}_1 \\ k_1\geq k_2\geq\cdots\geq k_n\geq 0}} \|K\|_{w,*} + \|\dot{p}\|_2^2 - |\mathrm{tr}(\dot{p}^*\dot{U}_1 K \dot{V}_1^*)| - |\mathrm{tr}(\dot{V}_1 K \dot{U}_1^* \dot{p})| + \|K\|_2^2 \\
\Leftrightarrow &\min_{\substack{\dot{U}_1,K,\dot{V}_1 \\ k_1\geq k_2\geq\cdots\geq k_n\geq 0}} \|K\|_{w,*} - |\mathrm{tr}(s\dot{U}_1^*\dot{U}k\dot{V}^*\dot{V}_1)| - |\mathrm{tr}(K\dot{U}^*\dot{U}_1 S\dot{V}_1^*\dot{V})| + \|K\|_2^2 \\
\Leftrightarrow &\min_{\substack{K \\ k_1\geq k_2\geq\cdots\geq k_n\geq 0}} \|K\|_{w,*} - \max_{\substack{(\dot{U}_1\dot{U})^*(\dot{U}_1\dot{U})=\dot{I} \\ (\dot{V}\dot{V}_1)^*(\dot{V}\dot{V}_1)=\dot{I}}} \left(|\mathrm{tr}(S(\dot{U}_1^*\dot{U})K(\dot{V}^*\dot{V}_1))| + |\mathrm{tr}(K(\dot{U}^*\dot{U}_1)S(\dot{V}_1^*\dot{V}))|\right) + \|K\|_2^2 \\
\Leftrightarrow &\min_{k_1\geq k_2\geq\cdots\geq k_n\geq 0} \|K\|_{w,*} - \sum_{i=1}^{n}\left(s_i k_i + k_i s_i\right) + \|k\|_2^2 \\
\Leftrightarrow &\min_{k_1\geq k_2\geq\cdots\geq k_n\geq 0} \sum_{i=1}^{n} w_i k_i - 2k_i s_i + k_i^2 \\
\Leftrightarrow &\min_{k_1\geq k_2\geq\cdots\geq k_n\geq 0} \sum_{i=1}^{n} (k_i - s_i)^2 + w_i k_i.
\end{aligned}
\tag{25}
$$

From the above deduction, we can see that the solution of the $\dot{g}$-subproblem is

$$\dot{g} = \dot{U}K\dot{V}^*. \tag{26}$$

□

Then the minimization problem (12) has a unique solution and can be solved by the ADMM method. We can compute the solution of the $\dot{u}$ subproblem according to Eq. (17). The process of our QWNNM algorithm is conclude in Algorithm 1. It is worth to mention that the blurring matrix $\dot{A}$ and $\dot{A}^*$ in (17) are not easy to handle. We synthesize a degraded image by using the Matlab built-in blurring kernels and then generate the corresponding blurring matrix. In the real number domain, the blurring matrix A is a monochromatic matrix. Since a quaternion matrix consists of three imaginary parts, how to generate a quaternion blurring matrix to solve $\dot{u}$ is a big challenge. By filling the three imaginary parts of the quaternion with the same blurring matrix A, we define the quaternion blurring matrix $\dot{A}$ as $\dot{A} = Ai + Aj + Ak$. We synthesize a degraded image by using the Matlab built-in blurring kernels and then generate the corresponding blurring matrix. For example, we use "H=fspecial('motion',20,60)" to create 2-D linear motion filter in Matlab and use the "padPSF" function in Matlab to pad an array with zeros to make it has the same dimension as the input image. Here the blur matrix is in real domain, however, we handle the color image restoration task in quaternion. Therefore, we generate a quaternion blur matrix in quaternion domain as

Algorithm 1 The proposed QWNNM algorithm.

Input:
Let $\dot{u}^0 = \dot{f}$, $\dot{g}^0 = \dot{u}^0$, $\dot{\eta}^0 = 0$;
Set parameters λ, β and w_i;
Output:
The recovered image $\dot{u}^k$;
for $k = 0 : \mathrm{kMax}$ **do**
  Calculate u^{k+1} by Eq. (17);
  Calculate g^{k+1} by Eq. (26);
  Update $\dot{\eta}^{k+1} = \dot{\eta}^k + (\dot{u}^{k+1} - \dot{g}^{k+1})$;
  k = k + 1;
end for

$$\dot{A} = A_r i + A_g j + A_b = \begin{pmatrix} \dot{A}_{11}, \dot{A}_{12}, \cdots, \dot{A}_{1n} \\ \dot{A}_{21}, \dot{A}_{22}, \cdots, \dot{A}_{2n}, \\ \vdots \\ \dot{A}_{m1}, \dot{A}_{m2}, \cdots, \dot{A}_{mn}, \end{pmatrix}, \tag{27}$$

where $A_r = A_g = A_b = H$ and $\dot{A}_{ij} = A_{ij}^r i + A_{ij}^g j + A_{ij}^b k$ denotes a pixel of $\dot{A}$, here $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. The blur matrix $\dot{A}$ here is a 2-D quaternion matrix. After applying the fast quaternion Fourier transform [30], the quaternion blurring matrix $\dot{A}$ and $\dot{A}^*$ can be expressed. In this fashion, all the elements in (17) can be computed in the quaternion domain and we can obtain the restoration result. The flowchart of the proposed method is displayed in Fig. 2. We first generate degradation images with blur and noise. Given the blurring kernel, we generate a blurring matrix corresponding to the original image. By applying the blurring kernel and adding noise to the original image, we synthesize a degraded color image. To solve the proposed model and better preserve the correlation of color channels, we transform the color image and the blurring matrix into the quaternion domain. By implementing the ADMM method and solving the $\dot{u}$, $\dot{g}$, and $\dot{\eta}$ subproblems, the recovery result is obtained.

Jia et al. [17] proposed a quaternion-based color image restoration model, which extended the classic TV model into the HSV color space, named SV-TV. Here we give some discussions between ST-TV and our QWNNM. Their SV-TV showed great success in color image restoration tasks, such as Gaussian noise removal, Gaussian blur with Gaussian noise restoration. However, they did not directly use the quaternion to represent the color image. Instead, they convert the quaternion into the HSV color space with the equation

$$\begin{cases} c_h(x,y) = \tan^{-1}\left(\frac{|u(x,y) - \mu\nu u(x,y)\nu\mu|}{|u(x,y) - \nu u(x,y)\nu|}\right), \\ c_S(x,y) = \frac{1}{2}|u(x,y) + \mu u(x,y)\mu|, \\ c_v(x,y) = \frac{1}{2}|u(x,y) - \mu u(x,y)\mu|, \end{cases} \tag{28}$$

where $u(x,y) = u_0(x,y) + u_1(x,y)i + u_2(x,y)j + u_3(x,y)k$ is the color image, u_t are the real and imaginary parts of $u(x,y), t = 0, 1, 2, 3$. Here μ referring to the gray-value axis, ν is a unit

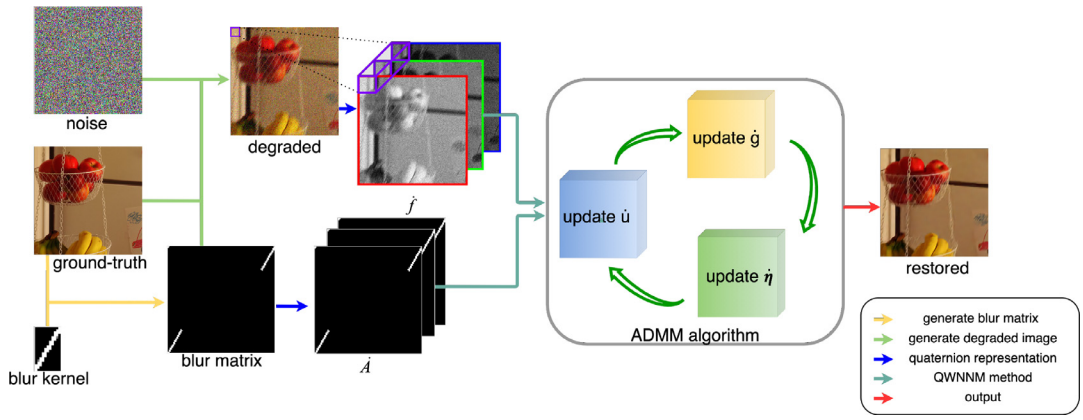


Fig. 2. The flowchart of the proposed method.

and pure quaternion number and ν is orthogonal to μ, and $c_h(x,y), c_s(x,y), c_v(x,y)$ are the HSV components, respectively. The main difference between the proposed method and SV-TV is the regularizer. While [17] utilized the SV-TV regularizer, the proposed method is based on the low rank regularizer, i.e., the effective WNNM regularizer. Due to the color information and structure, we extended the WNNM into the quaternion domain. Meanwhile, we studied the feasibility and the effectiveness of the proposed QWNNM strategy. According to the strategy proposed in [17], they did not easily transform the RGB color space into the HSV color space, but creatively from Eq. (28). Besides, they claimed that the H-component does not contain a lot of image details like edges and developed a TV regularization in S and V channels in the HSV color space. Such a creative scheme generates great success in handling color images corrupted by the Gaussian blur and noise. Therefore, it is not fair to compare SV-TV with our work by directly transforming our QWNNM into HSV color space. Here, we leave this work in the future.

4. Experiments

To demonstrate the effectiveness of the proposed model, we compare our method with several state-of-the-art methods for color image restoration, including LRTV [37], BM3D [40], SV-TV [17], MSWNNM [38], and the CNN-based method IRCNN [35]. We test with the images from Set27[1] and Set12[2], which are shown in Figs. 3 and 4, respectively. We test on three types of blur, "Gaussian", "motion", and "average", for color image deblurring. We use the Matlab command "fspecial('gaussian',25,1.6)", "fspecial('motion',20,60)", and "fspecial('average',9)" to generate the blurring kernels and add Gaussian white noise with Gaussian noise level (standard variance) $\sigma = 25$ to all images. In Matlab code, the fspecial ('motion', LEN,THETA) returns a filter to approximate, once convolved with an image, the linear motion of a camera by LEN pixels, with an angle of THETA degrees in a counter-clockwise direction. The filter becomes a vector for horizontal and vertical motions.

4.1. Parameter setting

The parameters of the proposed model are: $\lambda = 60$, $\beta = 7.5$ for Gaussian blur with noise; $\lambda = 110$, $\beta = 7.5$ for motion blur with noise; $\lambda = 110$, $\beta = 8.5$ for average blur with noise. All compared methods are implemented with the available code and the default parameters mentioned in their corresponding paper. All the experiments are conducted under Windows10 and MATLAB R2019a running on a desktop (Intel(R) Core(TM) i9-9900 CPU @3.60 GHz and 16GB memory). IRCNN was proposed by Zhang et. al. [35] to tackle image restoration tasks, such as image denoising, deblurring, and super-resolution. In this paper, we focus on the image restoration task. Since IRCNN achieved very promising restoration results, we set the IRCNN method as one of our benchmarks. In this paper, the IRCNN network was implemented with MatConvNet toolbox and an NVIDIA GeForce RTX 2080Ti. It is worth stating that we did not retrain the IRCNN[3]. It is a coincidence that the test images of our method are mostly the same as the test set of IRCNN. IRCNN method performs well in Gaussian blur removal, however, since the type of blur is not only a standard Gaussian distribution but also motion and average distribution, the restored image will be undesirable. The termination criterion is defined as

$$\frac{\|\dot{u}^{i+1} - \dot{u}^{i}\|}{\|\dot{u}^{i+1}\|} < 10^{-4}, \tag{29}$$

where i is the number of iterations. After we get the result $\dot{u}$, we transform it into the real space as $u^\star$ and evaluate the quality. The peak signal to noise ratio (PSNR) and structural similarity (SSIM) [41] are chosen as the quantitative measure. They are defined by

$$\mathrm{PSNR} = 20\log_{10}\frac{255}{\frac{1}{mn}\|u^\star - u\|_2}, \tag{30}$$

and

$$\mathrm{SSIM}(u, u^\star) = \frac{(2\mu_u\mu_{u^\star} + C_1)(2\sigma_{uu^\star} + C_2)}{\left(\mu_u^2 + \mu_{u^\star}^2 + C_1\right)\left(\sigma_u^2 + \sigma_{u^\star}^2 + C_2\right)}, \tag{31}$$

where u is the reference, μ_u, $\mu_u^\star$ and σ_u, $\sigma_u^\star$ are the mean and the standard deviation of u and $u^\star$, respectively. The positive constants C_1 and C_2 are used to avoid a null denominator.

In the experiments, we use the Matlab build-in function "psnr" and "ssim" to compute the PSNR and SSIM results. The higher the

[1] https://github.com/Huang-chao-yan/dataset. Image 1–7 and 16 are with the size of 500×500; Image 8 is with the size of 560×392; Image 9 and 22–25 are with the size of 481×321; Image 10 and 21 are with the size of 1024×683; Image 11 is with the size of 1024×764; Image 12 is with the size of 321×481; Image 13 is with the size of 1024×701; Image 14 is with the size of 1024×783; Image 15 is with the size of 1024×684; Image 17 is with the size of 1024×605; Image 18 is with the size of 448×296; Image 19 is with the size of 256×256; Image 20 is with the size of 288×288; Image 26 is with the size of 418×378; Image 27 is with the size of 1024×937.

[2] The images of Set12 are mainly collected from Wang et al. [22]. Bird, Plane, Baboon, Bee, Aquatic, Barbara, Boat, House, Peppers, and Starfish are with the size of 256×256, Lena and Pelican are with the size of 512×512.

[3] The code of IRCNN was downloaded from https://github.com/cszn/ircnn.

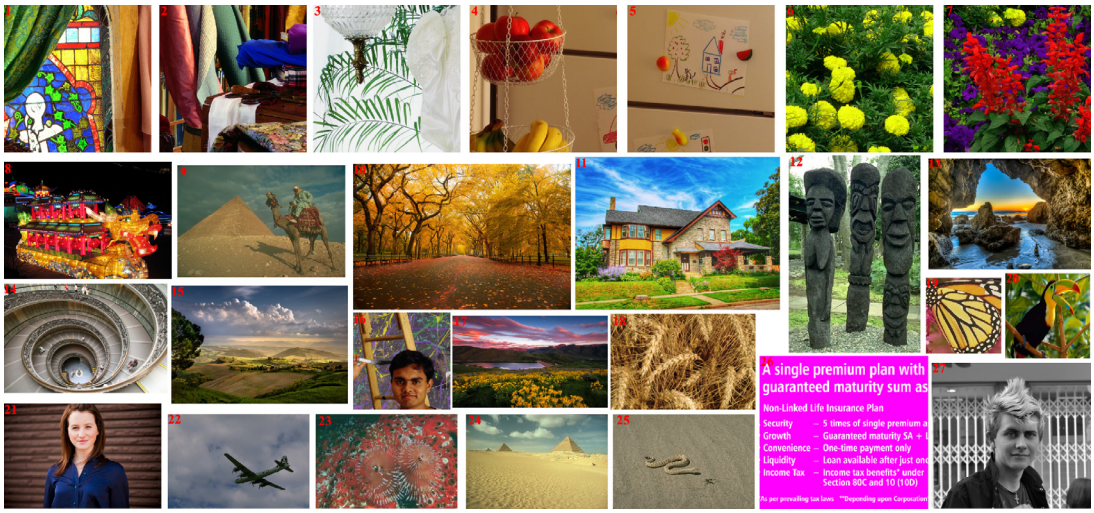
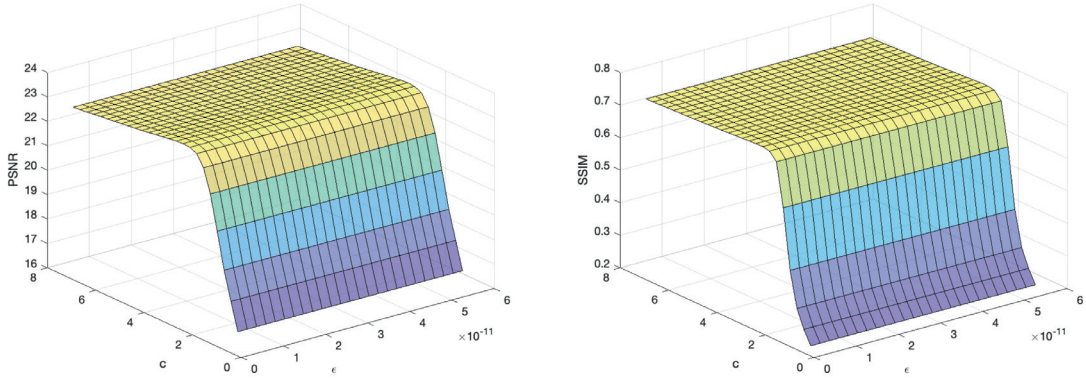


Fig. 3. Images in Set27.

Fig. 4. Images in Set12.



(a) PSNR values along with parameters c and ϵ.

(b) SSIM values along with parameters c and ϵ.

Fig. 5. Parameter analysis of image 'Plane' under the motion blur and noise.

PSNR and SSIM are, the better the quality of the restored image is, and the SSIM value is between 0 and 1. The weight $w_i = \frac{c}{\sigma_i + \epsilon}$, where ϵ is a small number and c is a constant. We analyze the effectiveness of different ϵ and c. From Fig. 5, we can see the PSNR and SSIM curves of image 'Plane' with motion blur and noise along with ϵ and c. Here, ϵ is a small constant, we set it with the Matlab built-in function 'eps' as $\epsilon \in [1 : 250000] * \text{eps}$ with step size 10000. The parameter c in WNNM [25] was set to be $\sqrt{2}$, here we set $c \in [0.1 : 5] * \sqrt{2}$ with step size 0.2. As shown in Fig. 5, the parameter ϵ is a small positive constant to avoid the denominator as zero. Hence, ϵ does not effectively affect the results, we set $\epsilon = \text{eps}$ for all our experiments. Meanwhile, c is a significant parameter in the weight w_i. The choice of c is directly affecting the weight w_i

Table 2
PSNR (dB) and SSIM values of different restoration models for GB(25,1.6)/σ=25.

Methods	Degraded	LRTV	BM3D	SV-TV	MSWNNM	IRCNN	QWNNM
Set27	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM
1	18.30 0.2725	22.97 0.6122	22.37 0.5959	21.89 0.5790	23.55 0.6569	23.75 0.6732	**25.33 0.7739**
2	19.28 0.2165	25.95 0.6471	25.78 0.6991	26.20 0.6949	26.93 0.7351	27.63 0.7465	**27.87 0.7519**
3	18.25 0.2951	23.55 0.7286	22.84 0.7757	23.44 0.8001	24.69 0.8327	26.03 0.8535	**26.53 0.8597**
4	19.52 0.1705	26.63 0.6569	26.79 0.7798	26.80 0.7709	27.86 0.8090	28.68 0.8172	**28.79 0.8259**
5	19.91 0.1423	29.08 0.6796	30.86 0.8641	30.26 0.8441	31.77 0.8699	32.50 0.8643	**32.73 0.8783**
6	18.97 0.2492	24.13 0.5984	23.57 0.5608	23.68 0.5614	24.64 0.6445	25.54 0.8121	**26.26 0.8774**
7	19.32 0.2518	25.49 0.6559	24.98 0.6506	24.32 0.6222	26.23 0.7109	26.83 0.9106	**27.37 0.9118**
8	18.24 0.2209	22.26 0.5897	22.03 0.6839	21.48 0.5740	23.17 0.7331	23.44 0.7376	**23.68 0.7464**
9	19.75 0.1347	27.64 0.6026	28.78 0.7190	28.96 0.7239	28.70 0.7143	29.31 0.7191	**29.51 0.9437**
10	17.59 0.2021	20.52 0.4138	19.86 0.3525	19.98 0.3688	**22.58 0.9112**	21.24 0.8826	21.68 0.8921
11	18.01 0.2183	21.59 0.5463	20.90 0.5435	21.01 0.5499	23.55 **0.8787**	25.36 0.7741	**25.44** 0.8582
12	18.06 0.2234	21.63 0.4538	21.49 0.4226	22.07 0.4751	21.92 0.4753	22.36 0.5195	**22.65 0.6347**
13	19.07 0.2218	24.47 0.6185	23.60 0.6000	24.05 0.6156	24.38 0.8064	25.89 0.7368	**25.99 0.8595**
14	18.73 0.2462	23.89 0.6685	22.82 0.6689	23.17 0.6814	23.59 0.7365	25.36 0.7741	**25.48 0.8383**
15	19.72 0.1529	27.52 0.6517	27.81 0.7485	28.10 0.7540	28.35 0.9046	29.41 0.7878	**29.87 0.9297**
16	19.69 0.2296	27.63 0.7119	27.31 0.7340	26.90 0.7130	28.51 0.7756	29.26 0.7929	**29.98 0.8068**
17	18.92 0.1774	24.02 0.6098	23.69 0.6653	23.74 0.6570	**25.69 0.8783**	25.09 0.7167	25.64 0.8764
18	17.17 0.2856	19.59 0.4317	19.35 0.3750	19.65 0.4388	19.81 0.4647	20.55 **0.5596**	**20.63** 0.5506
19	17.80 0.3594	22.68 0.7382	22.71 0.7759	23.90 0.8106	23.92 0.8151	24.96 **0.8385**	**25.13** 0.8282
20	19.57 0.2608	27.42 0.7415	27.79 0.8047	27.01 0.7528	28.69 0.8255	29.67 0.8390	**30.35 0.8566**
21	20.04 0.1396	30.35 0.7351	32.27 0.8817	32.51 0.8720	31.92 0.9517	33.76 0.8844	**34.29 0.9726**
22	20.01 0.1216	30.04 0.7123	33.49 0.9460	33.95 0.9282	34.16 0.9320	34.74 0.9103	**34.94 0.9595**
23	19.06 0.2477	24.42 0.5953	23.92 0.5616	24.43 0.6055	24.76 0.6275	25.69 0.6830	**25.98 0.8899**
24	19.79 0.1386	28.12 0.6109	29.19 0.6881	19.38 0.6945	29.30 0.6959	29.97 0.7689	**30.27 0.9674**
25	19.31 0.1187	25.30 0.3814	25.98 0.3957	26.01 0.4061	25.99 0.4034	26.28 0.4372	**26.46 0.7888**
26	17.23 0.1899	21.64 0.7592	20.38 0.8714	19.10 0.6785	19.70 0.9384	22.19 0.8326	**22.92 0.9718**
27	19.85 0.5849	29.13 0.8129	29.46 **0.9121**	29.48 0.9045	29.04 0.8212	32.14 0.8897	**32.32** 0.9032
Average	18.93 0.2249	25.10 0.6283	25.19 0.6769	24.87 0.6695	26.05 0.7611	26.95 0.7690	**27.34 0.8501**
Set12	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM
Bird	18.74 0.3797	23.34 0.7079	23.45 0.7234	23.82 0.7411	23.82 0.6556	24.32 0.6784	**24.63 0.7817**
Plane	18.99 0.3005	24.40 0.6892	24.82 0.8122	25.00 0.7907	25.36 0.7807	25.94 0.8013	**26.18 0.8440**
Baboon	18.39 0.5197	22.37 0.7042	22.33 0.6944	22.21 0.7026	22.56 0.5210	22.80 0.5223	**23.04 0.7321**
Bee	19.63 0.6109	28.02 0.8741	29.15 0.9167	27.64 0.8974	29.62 0.8371	29.52 0.8009	**30.40 0.9275**
Aquatic	18.67 0.3442	23.87 0.7262	24.14 0.7796	24.36 0.7789	24.64 0.7759	25.22 0.7788	**25.52 0.8287**
Barbara	18.56 0.5078	23.19 0.7862	23.37 0.7999	23.53 0.8072	23.83 0.6947	24.14 0.7134	**24.35 0.8339**
Boat	18.28 0.4001	22.28 0.7287	22.53 0.7865	22.79 0.7820	23.08 0.6725	**23.78** 0.7022	23.61 **0.8191**
House	19.31 0.6072	26.91 0.9084	27.54 0.9262	26.73 0.9142	**28.83** 0.7834	28.03 0.7770	28.49 **0.9376**
Peppers	18.86 0.7863	24.30 0.9367	24.84 0.9450	24.12 0.9342	25.62 0.7698	25.57 0.7826	**25.76 0.9539**
Starfish	18.85 0.7219	24.05 0.9199	24.14 0.9358	24.38 0.9347	24.64 0.7130	25.39 0.7419	**25.76 0.9512**
Lena	19.66 0.7764	27.58 0.9560	27.90 0.9609	28.11 0.9616	28.82 0.7634	29.23 0.7719	**29.57 0.9715**
Pelican	18.93 0.3630	24.53 0.7772	24.45 0.8208	25.00 0.8164	25.07 0.7291	**25.58** 0.7384	25.55 **0.8345**
Average	18.91 0.5265	24.57 0.8096	24.89 0.8418	24.81 0.8384	25.49 0.7222	25.79 0.7339	**26.07 0.8680**

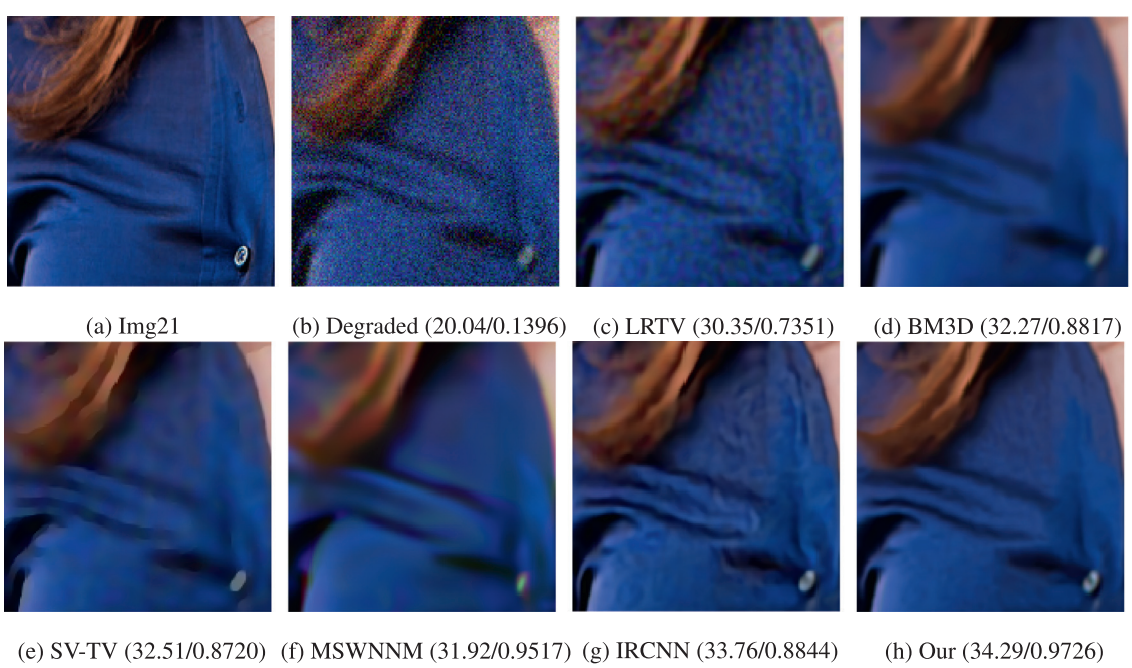


Fig. 6. Zoom in part of color image restoration on Img21 with visual quality and numerical results (PSNR/SSIM). (a) original image, (b) degraded image with Gaussian kernel (25, 1.6) and Gaussian noise level $\sigma = 25$, restored image reconstructed by: (c) LRTV [37], (d) BM3D [40], (e) SV-TV [17], (f) MSWNNM [38], (g) IRCNN [35], (h) our QWNNM.

Table 3
PSNR (dB) and SSIM values of different restoration models for MB(20,60)/σ=25.

Methods	Degraded	LRTV	BM3D	SV-TV	MSWNNM	IRCNN	QWNNM
Set27	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM
1	16.18 0.1493	20.42 0.5277	19.38 0.4720	19.59 0.4985	21.30 0.5798	21.22 0.7735	**21.57 0.8469**
2	17.89 0.1315	23.85 0.5962	22.98 0.6076	23.86 0.6268	24.70 0.6729	24.74 0.6698	**24.90 0.8358**
3	15.58 0.1585	19.94 0.6392	17.63 0.5784	19.27 0.6820	21.06 0.7364	21.57 0.7571	**22.39 0.7803**
4	18.81 0.1163	25.44 0.6536	24.67 0.7153	25.33 0.7405	26.02 0.7737	26.41 0.8804	**26.79 0.9612**
5	19.34 0.1119	27.85 0.6869	28.50 0.8389	28.14 0.8202	29.05 0.8427	29.48 0.9466	**29.81 0.9731**
6	17.34 0.1454	21.84 0.4714	21.36 0.4703	21.56 0.4362	22.19 0.4996	22.61 0.7206	**22.80 0.8548**
7	17.57 0.1493	22.40 0.5259	22.43 0.5448	21.52 0.4996	23.04 0.5777	22.91 0.7702	**23.33 0.8481**
8	16.23 0.1146	19.66 0.4825	20.40 0.6196	19.30 0.4781	20.57 0.6229	20.46 0.8103	**21.06 0.8102**
9	19.34 0.1048	26.85 0.5862	27.61 0.6926	27.45 0.6842	27.59 0.6878	27.84 0.8901	**27.93 0.9228**
10	16.32 0.1168	19.23 0.3355	18.90 0.3208	18.90 0.3104	**20.85 0.8871**	19.62 0.7626	19.94 0.8554
11	16.54 0.1289	20.07 0.4591	20.22 0.5464	19.51 0.4639	**21.14 0.8378**	20.51 0.7250	20.89 0.8236
12	16.94 0.1308	20.48 0.3667	20.24 0.3544	20.83 0.3765	20.73 0.3718	21.08 0.4087	**21.11 0.5486**
13	17.84 0.1375	22.56 0.5248	22.09 0.5407	22.14 0.5241	22.21 0.7580	23.31 0.7058	**23.52 0.7891**
14	17.11 0.1483	21.38 0.5629	20.97 0.5839	20.71 0.5641	21.05 0.6610	22.08 **0.7401**	**22.74** 0.7350
15	19.00 0.1120	25.90 0.6178	26.99 0.7395	15.93 0.7026	25.64 0.8755	26.92 0.8351	**27.05 0.8864**
16	18.60 0.1422	25.03 0.6230	24.03 0.6223	24.49 0.6233	25.62 0.6811	25.63 0.7806	**25.99 0.8057**
17	17.77 0.1189	22.45 0.5546	22.84 0.6643	22.08 0.5951	23.01 0.8148	22.97 0.7356	**23.39 0.8221**
18	15.92 0.1582	18.40 0.3350	17.84 0.2358	18.46 0.3317	18.41 0.3298	18.85 0.7921	**19.18 0.8255**
19	13.92 0.1373	17.72 0.5188	18.41 0.5962	18.81 0.6170	19.32 0.6297	19.77 0.7450	**20.52 0.8597**
20	17.94 0.1530	23.61 0.6384	23.07 0.6251	23.63 0.6368	24.85 0.7207	25.20 0.8321	**25.37 0.9157**
21	19.74 0.1217	29.74 0.7386	29.92 0.8607	30.66 0.8486	28.68 0.9164	**32.06** 0.8795	31.68 **0.9488**
22	19.49 0.1051	29.26 0.7420	31.36 0.9309	31.44 0.9253	**32.07 0.9383**	31.60 0.9298	31.75 0.9255
23	17.79 0.1437	22.19 0.4771	21.81 0.4529	22.23 0.4872	22.43 0.5015	22.71 0.7230	**23.12 0.8102**
24	19.42 0.1115	27.53 0.5827	28.60 0.6719	28.27 0.6559	28.40 0.6626	**28.66** 0.7674	28.52 **0.9511**
25	19.01 0.1013	25.02 0.3732	25.81 0.3899	25.35 0.3866	25.67 0.3891	25.67 0.5927	**25.83 0.7751**
26	14.59 0.0977	18.61 0.6956	17.54 0.8003	17.34 0.6386	20.10 0.9139	19.66 0.8282	**20.34 0.9562**
27	18.92 0.4576	27.16 0.8077	24.16 0.7765	25.95 0.8297	26.01 0.7669	**28.90 0.8663**	28.47 0.8353
Average	17.60 0.1409	23.13 0.5601	22.95 0.6019	22.69 0.5920	23.77 0.6907	24.16 0.7729	**24.44 0.8482**
Set12	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM
Bird	17.50 0.2792	21.66 0.6251	21.77 0.6395	22.09 0.6524	21.96 0.5428	22.45 0.5661	**22.66 0.6922**
Plane	17.45 0.2226	22.09 0.6451	22.21 0.7264	22.57 0.7276	22.86 0.6942	23.16 0.7261	**23.65 0.7735**
Baboon	17.38 0.4528	21.18 0.6560	21.23 0.6507	21.03 0.6532	21.52 0.4058	**21.60** 0.4226	21.59 **0.6679**
Bee	18.38 0.5739	25.46 0.8507	25.23 0.8702	24.78 0.8573	26.68 0.7957	26.34 0.7604	**26.96 0.8895**
Aquatic	16.45 0.2240	21.29 0.6223	21.19 0.6527	21.90 0.6773	22.09 0.6700	22.49 0.6739	**22.71 0.7105**
Barbara	17.24 0.4345	21.54 0.7311	21.60 0.7403	21.86 0.7521	21.90 0.5887	22.34 0.6161	**22.41 0.7715**
Boat	16.41 0.3030	20.12 0.6660	20.15 0.6970	20.76 0.7167	20.98 0.5748	21.46 0.6020	**21.75 0.7525**
House	17.65 0.5213	24.10 0.8604	23.69 0.8590	24.12 0.8599	25.23 0.7219	25.01 0.7246	**26.08 0.9026**
Peppers	17.18 0.7287	21.98 0.9022	22.18 0.9086	22.02 0.9009	**23.18** 0.6897	23.07 0.7060	23.07 **0.9189**
Starfish	16.99 0.6766	21.19 0.8924	21.18 0.8997	21.73 0.9055	21.59 0.5950	22.15 0.6201	**22.61 0.9195**
Lena	18.99 0.7586	26.41 0.9487	26.07 0.9504	26.55 0.9522	27.11 0.7301	**27.56** 0.7384	27.27 **0.9584**
Pelican	17.65 0.3235	23.15 0.7607	22.39 0.7809	23.23 0.7902	23.74 0.7008	23.68 0.7016	**24.10 0.8051**
Average	17.34 0.4582	22.51 0.7634	22.41 0.7813	22.72 0.7871	23.24 0.6425	23.44 0.6548	**23.74 0.8135**

Fig. 7. Zoom in part of color image restoration on Img5 with visual quality and numerical results (PSNR/SSIM). (a) original image, (b) degraded image with Gaussian kernel (25, 1.6) and Gaussian noise level $\sigma = 25$, restored image reconstructed by: (c) LRTV [37], (d) BM3D [40], (e) SV-TV [17], (f) MSWNNM [38], (g) IRCNN [35], (h) our QWNNM.

Table 4
PSNR (dB) and SSIM values of different restoration models for AB(9,9)/σ=25.

Methods Set27	Degraded PSNR SSIM	LRTV PSNR SSIM	BM3D PSNR SSIM	SV-TV PSNR SSIM	MSWNNM PSNR SSIM	IRCNN PSNR SSIM	QWNNM PSNR SSIM
1	16.93 0.1658	21.35 0.8304	20.55 0.8093	20.35 0.7013	22.07 0.5787	21.89 0.7768	**22.33 0.8541**
2	18.45 0.1520	24.53 0.8099	24.08 0.8194	24.60 0.6425	25.32 0.6844	25.55 0.7829	**25.65 0.8405**
3	16.43 0.1721	20.61 0.6334	19.75 0.6624	20.14 0.6746	21.34 0.7220	21.84 0.7430	**23.06 0.7720**
4	18.96 0.1201	25.49 0.9446	25.35 0.9488	25.38 0.7767	26.04 0.7583	26.40 0.9659	**26.55 0.9577**
5	19.58 0.1142	28.20 0.9584	28.88 0.9659	28.57 0.8215	29.89 0.8510	**30.21** 0.8543	30.18 **0.9725**
6	17.92 0.1419	22.51 0.8409	22.09 0.8407	22.12 0.6537	22.75 0.5193	22.99 0.7307	**23.25 0.8641**
7	18.37 0.1599	23.63 0.8445	23.25 0.8400	22.70 0.5289	24.12 0.6086	24.04 0.8030	**24.31 0.8678**
8	16.96 0.1243	20.64 0.7381	20.33 0.7088	20.02 0.7010	21.29 0.6490	21.17 0.6361	**21.79 0.7817**
9	19.53 0.1059	27.27 0.8842	28.08 0.9162	28.18 0.7001	28.19 0.7016	**28.45** 0.7034	28.21 **0.9230**
10	16.61 0.1000	19.56 0.8374	19.13 0.8330	19.17 0.3137	19.17 **0.8901**	19.80 0.5607	**20.08** 0.8577
11	17.04 0.1249	20.66 0.4999	20.06 0.7931	20.11 0.4938	21.00 **0.8513**	20.98 0.7492	**21.33** 0.8326
12	17.22 0.1259	20.85 0.3835	20.71 0.5067	21.12 0.3866	21.05 0.3901	21.37 0.4137	**21.49 0.5549**
13	18.26 0.1355	23.18 0.5554	22.57 0.7643	22.68 0.5374	22.81 0.7717	23.69 0.6126	**23.85 0.7937**
14	17.69 0.1503	22.24 0.6028	21.36 0.6743	21.33 0.5775	21.62 0.6676	22.67 0.6579	**23.31 0.7420**
15	19.35 0.1164	26.76 0.6623	26.82 0.8939	26.92 0.7250	26.83 0.8921	**27.72** 0.7590	**27.72 0.8963**
16	18.99 0.1553	25.80 0.6496	25.40 0.7826	25.11 0.6354	26.31 0.6954	26.28 0.6961	**26.72 0.8162**
17	18.26 0.1233	23.12 0.5928	22.78 0.8207	22.79 0.6199	23.57 **0.8588**	23.55 0.6611	**23.75** 0.8290
18	16.01 0.1179	18.46 0.2889	18.30 0.7806	18.41 0.2763	18.43 0.2817	18.69 0.3142	**18.89 0.8040**
19	15.87 0.2209	20.12 0.6336	19.81 0.8281	20.78 0.7027	20.91 0.7016	21.14 0.7115	**22.21 0.8957**
20	18.68 0.1857	25.28 0.6773	25.32 0.9124	24.95 0.6754	26.13 0.7490	26.36 0.7570	**26.46 0.9258**
21	19.89 0.1255	30.14 0.7743	31.21 0.9459	31.42 0.8594	30.09 0.9394	**32.63** 0.8889	32.29 **0.9492**
22	19.82 0.1099	29.87 0.7886	31.64 0.9294	32.85 0.9354	32.83 0.9435	**32.95 0.9411**	32.84 0.9183
23	18.14 0.1385	22.75 0.4896	22.42 0.7791	22.54 0.4785	22.80 0.5016	23.03 0.7100	**23.34 0.8138**
24	19.57 0.1084	27.86 0.6174	28.59 0.9528	28.67 0.6648	28.83 0.6724	**29.11** 0.8766	29.00 **0.9538**
25	19.08 0.0879	25.08 0.3669	25.60 0.7679	25.58 0.3853	25.80 0.3884	**25.91** 0.6883	**25.91 0.7715**
26	15.23 0.1180	18.88 0.7715	18.18 0.9197	17.91 0.6713	19.21 0.9045	19.14 0.8409	**20.35 0.9510**
27	19.44 0.5368	28.26 0.8149	27.94 0.8345	27.35 0.8700	26.01 0.7669	**30.17 0.8814**	29.98 0.8335
Average	18.08 0.1495	23.82 0.6849	23.71 0.8234	23.77 0.6290	24.24 0.7014	24.73 0.7302	**24.99 0.8508**
Set12	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM	PSNR SSIM
Bird	17.90 0.2956	22.21 0.6460	22.25 0.6630	22.40 0.6656	22.34 0.6678	22.65 0.5665	**22.77 0.6886**
Plane	17.99 0.2282	22.87 0.6562	23.01 0.7473	23.26 0.7429	23.39 0.7618	23.82 0.7376	**24.15 0.7574**
Baboon	17.63 0.4520	21.52 0.6601	21.54 0.6609	21.32 0.6536	21.66 0.6661	21.82 0.4909	**21.90 0.6728**
Bee	18.98 0.5857	26.60 0.8609	27.02 0.8916	26.12 0.8773	27.78 0.8916	27.50 0.7801	**27.90 0.8936**
Aquatic	17.32 0.2475	22.09 0.6552	22.02 0.6891	22.39 0.6990	22.50 0.7140	23.01 0.6851	**23.35 0.7341**
Barbara	17.57 0.4359	21.89 0.7345	21.94 0.7455	22.10 0.7505	22.34 0.7603	22.64 0.6195	**22.85 0.7807**
Boat	17.13 0.3173	21.00 0.6840	20.96 0.7212	21.27 0.7254	21.35 0.7408	21.82 0.6082	**22.17 0.7530**
House	18.48 0.5530	25.61 0.8849	25.28 0.8862	25.29 0.8842	26.89 0.9081	26.17 0.7438	**27.64 0.9220**
Peppers	17.75 0.7414	22.64 0.9115	22.83 0.9168	22.35 0.9051	23.74 0.9293	23.57 0.7188	**23.87 0.9317**
Starfish	17.74 0.6878	22.27 0.9010	22.21 0.9128	22.31 0.9106	22.66 0.9194	22.91 0.6235	**23.27 0.9231**
Lena	19.15 0.7576	26.52 0.9477	26.39 0.9502	26.59 0.9502	27.29 0.9553	**27.55** 0.7285	27.51 **0.9578**
pelican	18.31 0.3316	23.75 0.7796	23.45 0.7951	23.92 0.7979	24.10 **0.8211**	24.30 0.7036	**24.38** 0.7879
Average	18.00 0.4695	23.25 0.7768	23.24 0.7983	23.28 0.7969	23.84 0.8113	23.98 0.6604	**24.31 0.8169**

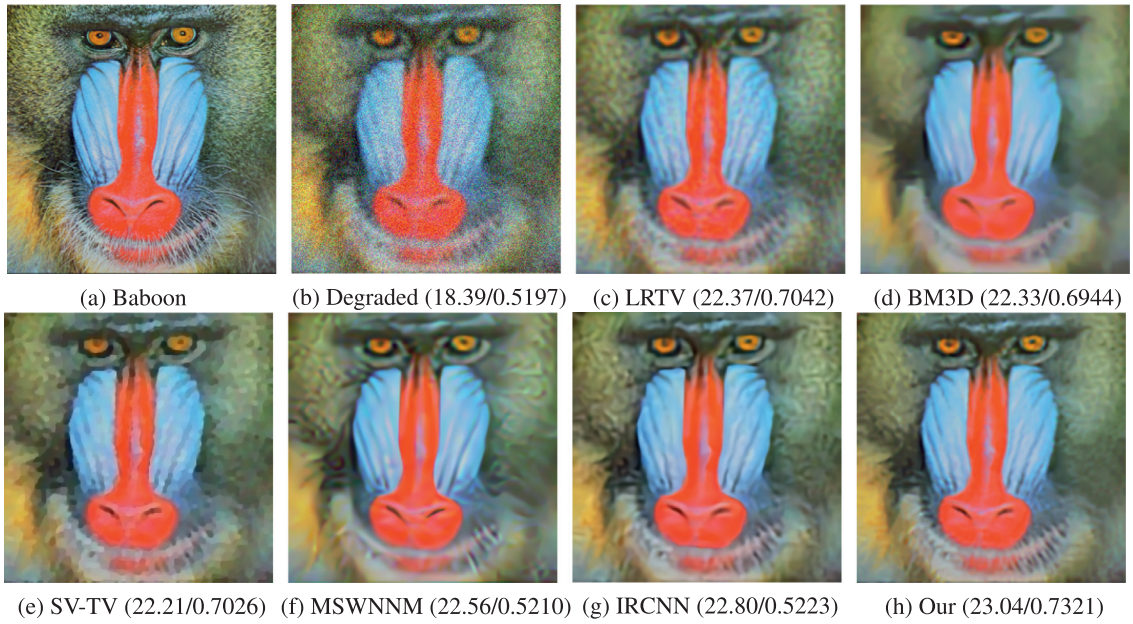


Fig. 8. Color image restoration on 'Baboon' with visual quality and numerical results (PSNR/SSIM). (a) original image, (b) degraded image with Gaussian kernel (25, 1.6) and Gaussian noise level $\sigma = 25$, restored image reconstructed by: (c) LRTV [37], (d) BM3D [40], (e) SV-TV [17], (f) MSWNNM [38], (g) IRCNN [35], (h) our QWNNM.

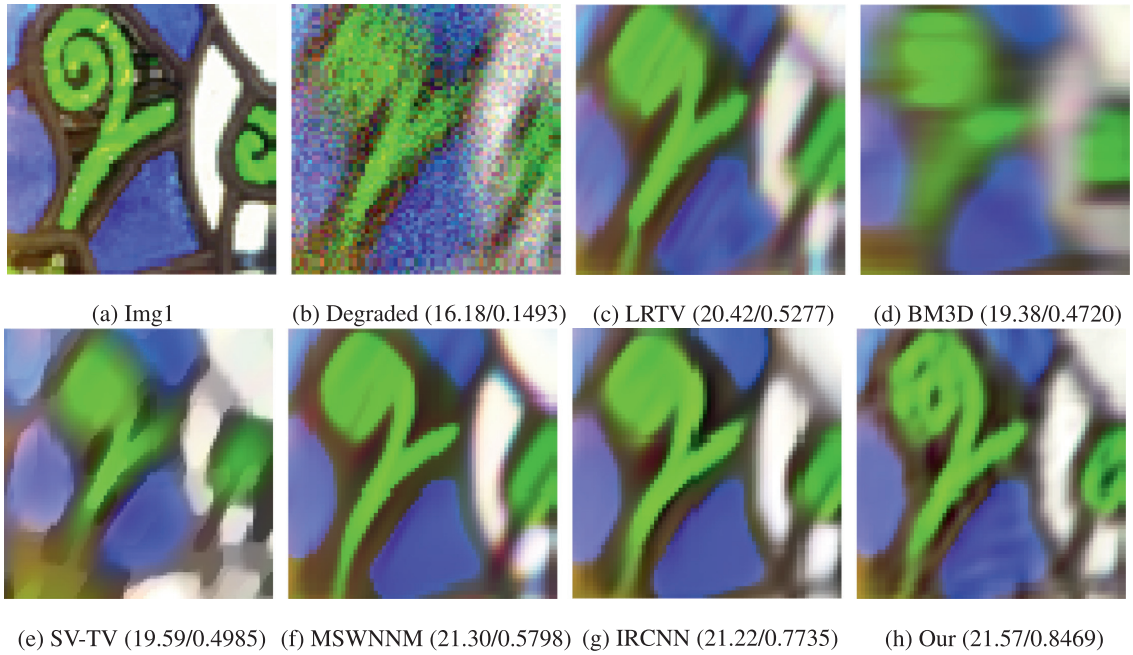


Fig. 9. Zoom in part of color image restoration on Img1 with visual quality and numerical results (PSNR/SSIM). (a) original image, (b) degraded image with motion kernel (20, 60) and Gaussian noise level $\sigma = 25$, restored image reconstructed by: (c) LRTV [37], (d) BM3D [40], (e) SV-TV [17], (f) MSWNNM [38], (g) IRCNN [35], (h) our QWNNM.

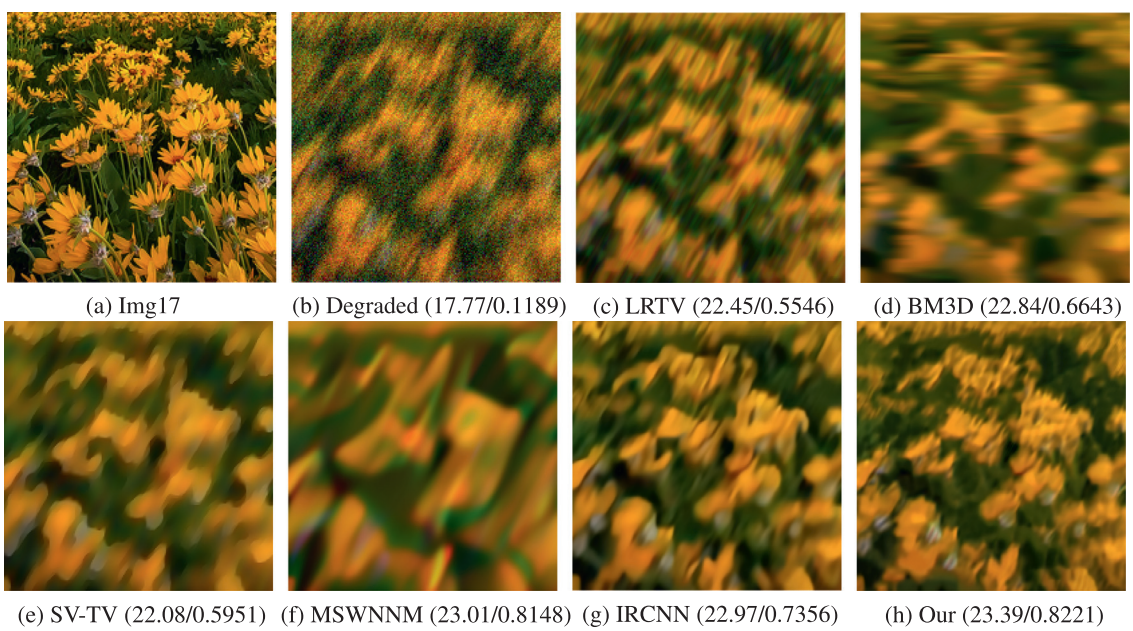


Fig. 10. Zoom in part of color image restoration on Img17 with visual quality and numerical results (PSNR/SSIM). (a) original image, (b) degraded image with motion kernel (20, 60) and Gaussian noise level $\sigma = 25$, restored image reconstructed by: (c) LRTV [37], (d) BM3D [40], (e) SV-TV [17], (f) MSWNNM [38], (g) IRCNN [35], (h) our QWNNM.

and the final result. Through trial and error, we set $c = 1.7 * \sqrt{2}$ for all our experiments.

4.2. Restoration results

The PSNR and SSIM values of compared restoration methods with Gaussian blur $GB(25, 1.6)/\sigma = 25$ is shown in Table 2, The corresponding results with motion blur $MB(20, 60)/\sigma = 25$ and with average blur $AB(9, 9)/\sigma = 25$ are shown in Tables 3 and 4, respectively. We highlight the best results in bold and underline the second-best results. From the numerical results, we see that the proposed method generates better restoration results with different types of blur. We also display the restoration images to demonstrate the visual quality of the proposed method. Figs. 6, 7 and 8 are the restoration results of images degraded by Gaussian blur and noise. The color spots are still visible in the results of LRTV [37] and MSWNNM [38], which are the WNNM-

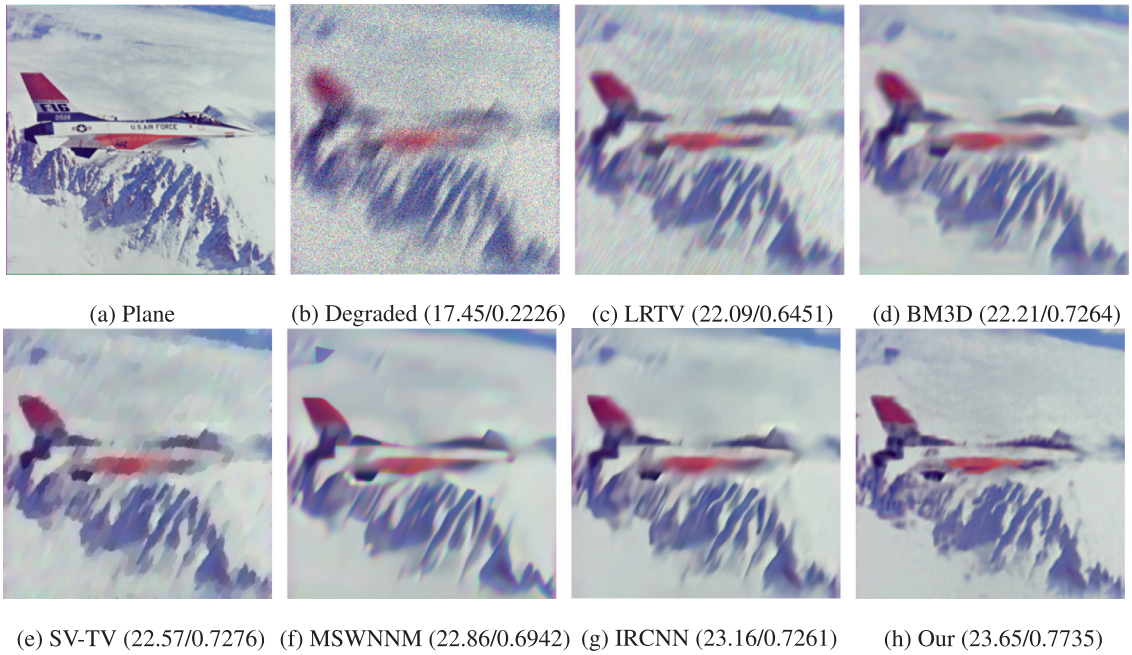


Fig. 11. Color image restoration on 'Plane' with visual quality and numerical results (PSNR/SSIM). (a) original image, (b) degraded image with motion kernel (20, 60) and Gaussian noise level $\sigma = 25$, restored image reconstructed by: (c) LRTV [37], (d) BM3D [40], (e) SV-TV [17], (f) MSWNNM [38], (g) IRCNN [35], (h) our QWNNM.

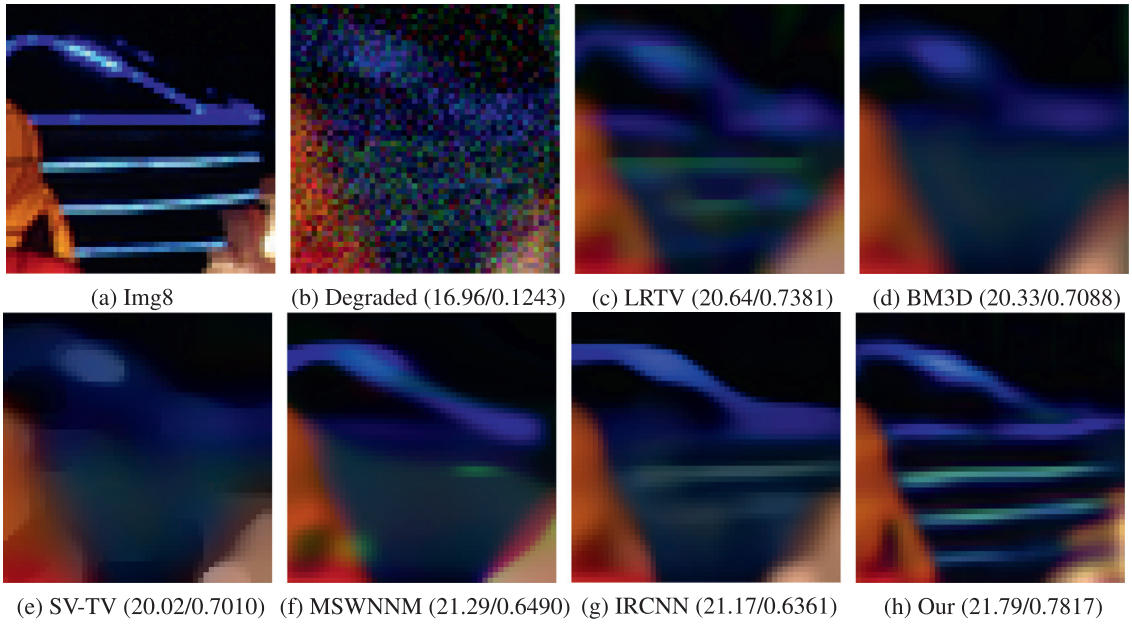


Fig. 12. Zoom in part of color image restoration on Img8 with visual quality and numerical results (PSNR/SSIM). (a) original image, (b) degraded image with average kernel (9, 9) and Gaussian noise level $\sigma = 25$, restored image reconstructed by: (c) LRTV [37], (d) BM3D [40], (e) SV-TV [17], (f) MSWNNM [38], (g) IRCNN [35], (h) our QWNNM.

based method in the real number domain. The blur seems not to be removed fully in the restored images of BM3D [40]. The SV-TV [17] is a quaternion-based method, which can generate good results in low Gaussian noise level. However, with Gaussian noise level $\sigma = 25$, the restoration results seem not so good. We also compared our method with the IRCNN [35], which is a convolutional neural network method. From the visual results, we observe that the proposed method can better preserve the color structure of the color channels. Figs. 9, 10, and 11 are the results of restoration with motion blur and noise. In Fig. 9, we see that the proposed method can better preserve the detailed information. For Fig. 10, the zoom-in part of the original image is a flower from the lower right corner to the upper left corner, and the degradation we added is motion blur moving to the right. For Fig. 11, even the mount below the plane is in the same direction with the motion blur, the restoration of our QWNNM still has the best result. It is not difficult to see from the degradation diagram that degradation is quite severe. Compared with the results of other methods, only our method can show the real shape of the image. Figs. 12, 13, and 14 are the restoration results of average blur with noise.

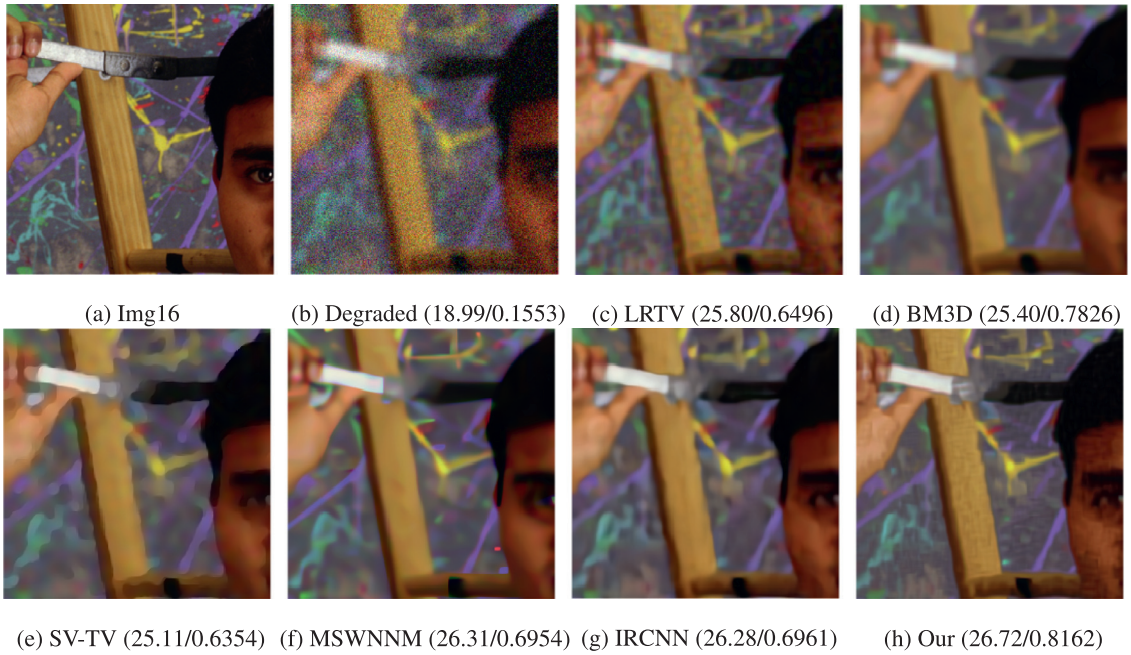


Fig. 13. Zoom in part of color image restoration on Img16 with visual quality and numerical results (PSNR/SSIM). (a) original image, (b) degraded image with average kernel (9, 9) and Gaussian noise level $\sigma = 25$, restored image reconstructed by: (c) LRTV [37], (d) BM3D [40], (e) SV-TV [17], (f) MSWNNM [38], (g) IRCNN [35], (h) our QWNNM.

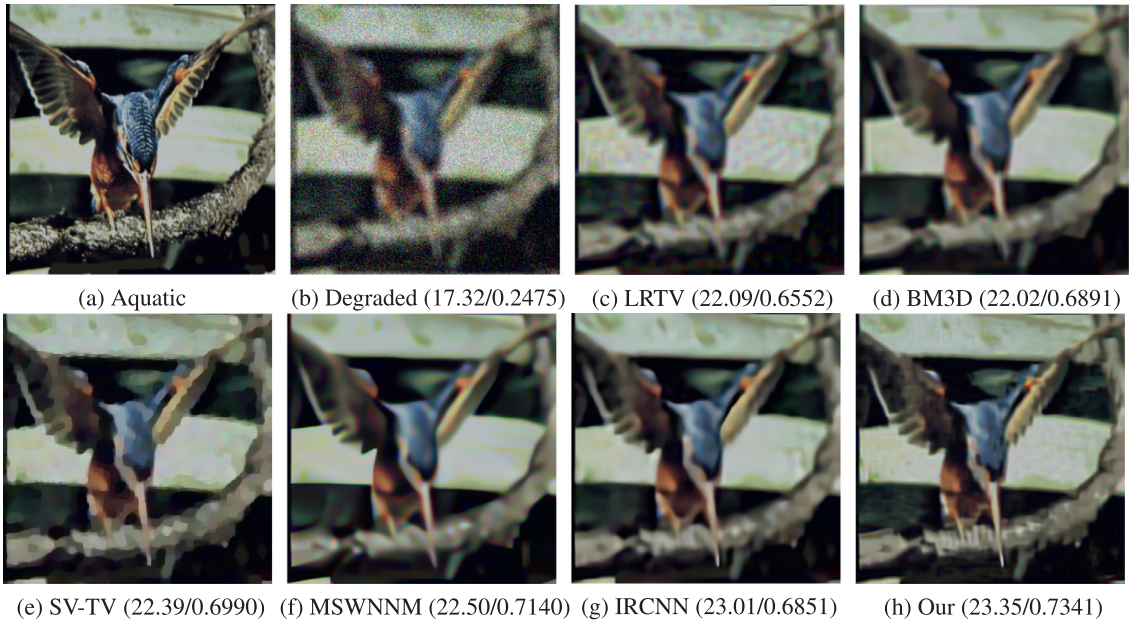
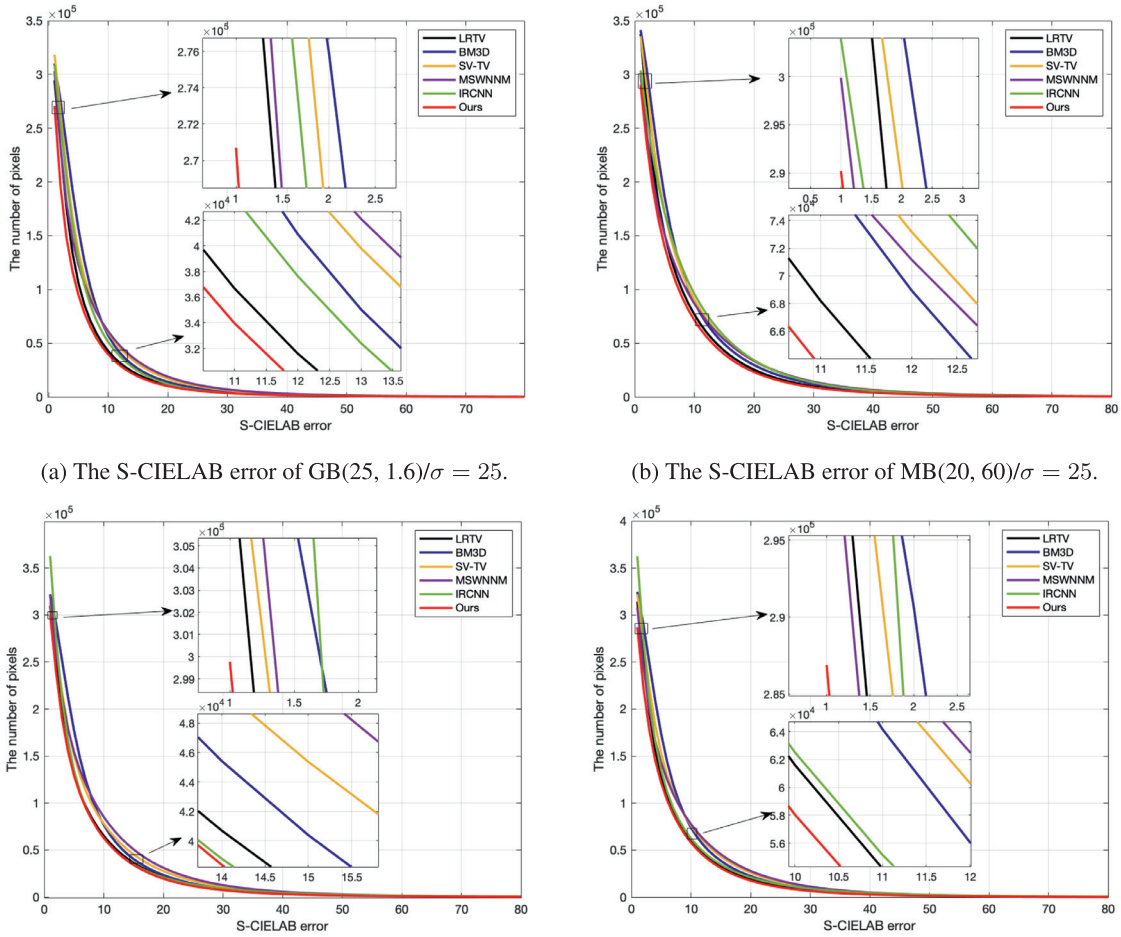


Fig. 14. Color image restoration on 'Aquatic' with visual quality and numerical results (PSNR/SSIM). (a) original image, (b) degraded image with average kernel (9, 9) and Gaussian noise level $\sigma = 25$, restored image reconstructed by: (c) LRTV [37], (d) BM3D [40], (e) SV-TV [17], (f) MSWNNM [38], (g) IRCNN [35], (h) our QWNNM.

From Fig. 12, we observe that the detailed structures of our resorted color images are better than the others. In the background of Fig. 13, we can see that the color distribution is avoided by our method. From Fig. 14, we know that compared with other restoration methods, our QWNNM can better prevent the oversmoothing and recover the detailed structure of the image. From all test images, we see that the proposed method has the best performance in both visual quality and numerical results.

To better demonstrate the advantages of the proposed method in color image restoration, we use the S-CIELAB error[4] to analyze the color quality of all compared methods in Fig. 15. We set the S-CIELAB error from 0 to 80 to study the number of error pixels between the original image and the restored image. The S-CIELAB error is used to evaluate the error between a pair of images that are presently coded as RGB data. We input the original image and the restored image as a pair and evaluate the S-CIELAB error from 0 to 80. We plot the S-CIELAB error of all the test images with the three different blur. The curves show that the proposed method has the least error pixels when the S-CIELAB error is zero and when the curves converge.

[4] http://scarlet.stanford.edu/brian/scielab/scielab.html.



(a) The S-CIELAB error of GB(25, 1.6)/$\sigma = 25$.

(b) The S-CIELAB error of MB(20, 60)/$\sigma = 25$.

(c) The S-CIELAB error of AB(9, 9)/$\sigma = 25$.

(d) The average S-CIELAB error.

Fig. 15. The S-CIELAB error of all compared methods on Set27. (a) the average S-CIELAB error of all degraded images with GB(25, 1.6)/$\sigma = 25$, (b) the average S-CIELAB error of all degraded images with MB(20, 60)/$\sigma = 25$, (c) the average S-CIELAB error of all degraded images with AB(9, 9)/$\sigma = 25$, (d) the average S-CIELAB error of all test images and all blur types.

5. Conclusions

In this paper, we proposed and analyzed a quaternion-based model for color image restoration, which can overcome the disadvantages of existing real-valued WNNM-based methods. We represented the color image with a pure quaternion matrix and applied the classical WNNM method in the quaternion domain. Hence, the correlation of color channels can be well preserved. Due to the special multiplication and derivation rules of the quaternion, the mathematical deduction of the weighted nuclear norm is different from the real-valued deduction. Therefore, we carefully designed a blurring operator in the quaternion domain and proved the uniqueness of the optimal solution. In three types of blur kernels with noise, our QWNNM method generated better restoration results, which demonstrate the robustness of the proposed model. Both visual and numerical results on two different datasets illustrated the robustness of the proposed scheme. Compared with both the real-valued traditional and convolutional neural network methods, our quaternion-based method can better preserve the color structure and avoid color distribution. However, the proposed strategy has its limitations. We applied the classical ADMM solver to handle our QWNNM model, in fact, there is a more accurate solution for the proposed convex problem. Later on, we plan to design a more advanced algorithm to improve the accuracy of the solution.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The research was in part by the National Key R&D Program of China under Grant 2021YFE0203700, Grant NSFC/RGC N_CUHK 415/19, Grant ITF MHP/038/20, Grant RGC 14300219, 14302920, 14301121, and CUHK Direct Grant for Research under Grant 4053405, 4053460, the Natural Science Foundation of China (Grant Nos. 61971234, 12126340, 12126304, 11501301, 62001167), the “1311 Talent Plan” of NUPT and the Science, the “QingLan” Project for Colleges, the Graduate Student Scientific Research Innovation Project of Jiangsu Province, under Grant KYCX20_0788.

References

[1] J. Duan, Z. Pan, B. Zhang, W. Liu, X. Tai, Fast algorithm for color texture image inpainting using the non-local CTV model, J. Global Optim. 62 (4) (2015) 853–876.

[2] J. Liu, W. Liu, S. Ma, M. Wang, L. Li, G. Chen, Image-set based face recognition using K-SVD dictionary learning, Int. J. Mach. Learn. Cybern. 10 (2019) 1051–1064.

[3] M. Zhang, G.S. Young, Y. Tie, X. Gu, X. Xu, A new framework of designing iterative techniques for image deblurring, Pattern Recognit. 124 (2022) 108463.
[4] Y. Ma, X. Liu, S. Bai, L. Wang, A. Liu, D. Tao, E. Hancock, Region-wise generative adversarial image inpainting for large missing areas, arXiv preprint arXiv:1909.12507 (2019).
[5] Q. Dai, F. Fang, J. Li, G. Zhang, A. Zhou, Edge-guided composition network for image stitching, Pattern Recognit. 118 (2021) 108019.
[6] C. Huang, M.K. Ng, T. Wu, T. Zeng, Quaternion-based dictionary learning and saturation-value total variation regularization for color image restoration, IEEE Trans. Multimedia (2021) 1–13.
[7] Z. Shao, H. Shu, J. Wu, B. Chen, J. Coatrieux, Quaternion bessel-Fourier moments and their invariant descriptors for object reconstruction and recognition, Pattern Recognit. 47 (2014) 603–611.
[8] W.R. Hamilton, Elements of Quaternions, Longmans, Green, & Company, 1866.
[9] Ö.N. Subakan, B.C. Vemuri, A quaternion framework for color image smoothing and segmentation, Int. J. Comput. Vis. 91 (3) (2011) 233–250.
[10] X. Li, Y. Zhou, J. Zhang, Quaternion non-local total variation for color image denoising, in: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), 2019, pp. 1602–1607.
[11] X. Li, Y. Zhou, J. Zhang, Color image denoising using quaternion adaptive non-local coupled means, in: IEEE International Conference on Image Processing (ICIP), 2019, pp. 1810–1814.
[12] Y. Chen, X. Xiao, Y. Zhou, Low-rank quaternion approximation for color image processing, IEEE Trans. Image Process. 29 (2019) 1426–1439.
[13] Y. Yu, Y. Zhang, S. Yuan, Quaternion-based weighted nuclear norm minimization for color image denoising, Neurocomputing 332 (2019) 283–297.
[14] J. Shi, X. Zheng, J. Wu, B. Gong, Q. Zhang, S. Ying, Quaternion Grassmann average network for learning representation of histopathological image, Pattern Recognit. 89 (2019) 67–76.
[15] X. Zhu, Y. Xu, H. Xu, C. Chen, Quaternion convolutional neural networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 631–647.
[16] Q. Yin, J. Wang, X. Luo, J. Zhai, S. Jha, Y. Shi, Quaternion convolutional neural network for color image classification and forensics, IEEE Access 7 (2019) 20293–20301.
[17] Z. Jia, M.K. Ng, W. Wang, Color image restoration by saturation-value total variation, SIAM J. Imaging Sci. 12 (2) (2019) 972–1000.
[18] K. Mei, J. Li, J. Zhang, H. Wu, J. Li, R. Huang, Higher-resolution network for image demosaicing and enhancing, IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019, pp. 3441–3448.
[19] J. Li, J. Li, F. Fang, F. Li, G. Zhang, Luminance-aware pyramid network for low–light image enhancement, IEEE Trans. Multimedia 23 (2020) 3153–3165.
[20] F. Liu, S. Wang, J. Qin, Y. Lou, J. Rosenberger, Estimating latent brain sources with low-rank representation and graph regularization, in: International Conference on Brain Informatics, 2018, pp. 304–316.
[21] X. Jiang, L. Zhang, L. Qiao, D. Shen, Estimating functional connectivity networks via low-rank tensor approximation with applications to MCI identification, IEEE Trans. Biomed. Eng. 67 (7) (2020) 1912–1920.
[22] F. Wang, H. Huang, J. Liu, Variational based mixed noise removal with CNN deep learning regularization, IEEE Trans. Image Process. 29 (1) (2020) 1246–1258.
[23] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, L. Zhang, Weighted nuclear norm minimization and its applications to low level vision, Int. J. Comput. Vis. (2016) 183–208.
[24] Z. Kang, C. Peng, Q. Cheng, Robust PCA via nonconvex rank approximation, in: Processing of IEEE International Conference, 2015, pp. 211–220.
[25] S. Gu, L. Zhang, W. Zuo, X. Feng, Weighted nuclear norm minimization with application to image denoising, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2862–2869.
[26] Z. Xia, X. Wang, W. Zhou, R. Li, C. Wang, C. Zhang, Color medical image lossless watermarking using chaotic system and accurate quaternion polar harmonic transforms, Signal Process. 157 (2019) 108–118.
[27] T. Yang, J. Ma, Y. Miao, X. Wang, B. Xiao, B. He, Q. Meng, Quaternion weighted spherical bessel-fourier moment and its invariant for color image reconstruction and object recognition, Inf. Sci. 505 (2019) 388–405.
[28] F. Zhang, Quaternions and matrices of quaternions, Linear Algebra Appl. 251 (1997) 21–57.
[29] X. Liu, Y. Chen, Z. Peng, J. Wu, Z. Wang, Infrared image super-resolution reconstruction based on quaternion fractional order total variation with Lp quasi-norm, Appl. Sci. 8 (10) (2018) 1864–1887.
[30] S.J. Sangwine, Fourier transforms of colour images using quaternion or hypercomplex, numbers, Electron. Lett. 32 (21) (1996) 1979–1980.
[31] D. Xu, D.P. Mandic, The theory of quaternion matrix derivatives, IEEE Trans. Signal Process. 63 (2015) 1543–1556.
[32] Z. Jia, S. Ling, M. Zhao, Color two-dimensional principal component analysis for face recognition based on quaternion model, in: International Conference on Intelligent Computing, 2017, pp. 177–189.
[33] Y. Chen, Z. Jia, Y. Peng, Y. Peng, Robust dual-color watermarking based on quaternion singular value decomposition, IEEE Access 8 (2020) 30628–30642.
[34] L. Qi, X. Zhang, Constrained optimization of real functions in quaternion matrix variables, arXiv preprint arXiv:2009.13884 (2020).
[35] K. Zhang, W. Zuo, S. Gu, L. Zhang, Learning deep CNN denoiser prior for image restoration, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3929–3938.
[36] Q. Xie, D. Meng, S. Gu, L. Zhang, W. Zuo, X. Feng, Z. Xu, On the optimal solution of weighted nuclear norm minimization, arXiv preprint arXiv:1405.6012 (2014).
[37] L. Ma, L. Xu, T. Zeng, Low rank prior and total variation regularization for image deblurring, J. Sci. Comput. (2017) 1336–1357.
[38] N. Yair, T. Michaeli, Multi-scale weighted nuclear norm image restoration, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 3165–3174.
[39] Z. Wu, Q. Wang, J. Jin, Y. Shen, Structure tensor total variation-regularized weighted nuclear norm minimization for hyperspectral image mixed denoising, Signal Process. 131 (2017) 202–219.
[40] Y.M. kinen, L. Azzari, A. Foi, Exact transform-domain noise variance for collaborative filtering of stationary correlated noise, in: IEEE International Conference on Image Processing, 2019, pp. 185–189.
[41] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.

Chaoyan Huang received the BS degree in College of Mathematics and Computer Science from the Anqing Normal University, Anqing, China, in 2019. She is currently pursuing the MS degree with the School of Science, Nanjing University of Posts and Telecommunications, Nanjing, China. Her research interests include image processing and machine learning.

Zhi Li received the BS degree and the MS degree from China University of Petroleum, Shandong, China, in 2007 and 2010, respectively. He also received the MS degree in applied science from Saint Maryś University, Halifax, NS, Canada, in 2012. After being awarded the Hong Kong PhD Fellowship, he went to Hong Kong Baptist University, Hong Kong, where he received the PhD degree in 2016. Then he worked as a Postdoctoral Researcher at Michigan State University, East Lansing, MI, USA, from 2016 to 2019. He is currently an associate researcher with the Department of Computer Science and Technology, East China Normal University, Shanghai, China.

Yubing Liu is currently pursuing his BS degree with Bell Honor School, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include image processing, computer vision, and machine learning.

Tingting Wu is currently an Associate Professor in the School of Science, Nanjing University of Posts and Telecommunications, Nanjing, China. She received the BS and PhD degrees in mathematics from Hunan University, Changsha, China, in 2006 and 2011, respectively. In 2015–2018, she was a Post-Doctoral Researcher with the School of Mathematical Sciences, Nanjing Normal University, Nanjing, China. In 2016–2017, she was a Research Fellow in Nanyang Technological University, Singapore. Her research interests include variational methods for image processing and computer vision, optimization methods and their applications in sparse recovery and regularized inverse problems.

Tieyong Zeng is currently a Professor in the Department of Mathematics, The Chinese University of Hong Kong. He received the BS degree from Peking University, Beijing, China, the MS degree from Ecole Polytechnique, Palaiseau, France, and the PhD degree from the University of Paris XIII, Paris, France, in 2000, 2004, and 2007, respectively. His research interests include image processing, optimization, artificial intelligence, scientific computing, computer vision, machine learning, and inverse problems.