INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 19, Number 5, Pages 709–737

TOTAL VARIATION BASED PURE QUATERNION DICTIONARY LEARNING METHOD FOR COLOR IMAGE DENOISING

TINGTING WU, CHAOYAN HUANG, ZHENGMENG JIN, ZHIGANG JIA, AND MICHAEL K. NG

Abstract. As an important pre-processing step for many related computer vision tasks, color image denoising has attracted considerable attention in image processing. However, traditional methods often regard the red, green, and blue channels of color images independently without considering the correlations among the three channels. In order to overcome this deficiency, this paper proposes a novel dictionary method for color image denoising based on pure quaternion representation, which efficiently deals with both single-channel and cross-channel information. The pure quaternion constraint is firstly used to force the sparse representations of color images to contain only red, green, and blue color information. Moreover, a total variation regularization is proposed in the quaternion domain and embedded into the pure quaternion-based representation model, which is effective to recover the sharp edges of color images. To solve the proposed model, a new numerical scheme is also developed based on the alternating minimization method (AMM). Experimental results demonstrate that the proposed model has better denoising results than the state-of-the-art methods, including a deep learning approach DnCNN, in terms of PSNR, SSIM, and visual quality.

Key words. Color image denoising, singular value decomposition, pure quaternion matrix, total variation, sparse representation.

1. Introduction

Color image denoising is a fundamental image processing task that focuses on obtaining a clean color image from a noisy observation [39]. Color images have been widely used in many fields, from medical imaging to automatic driving [15, 47, 53] Generally speaking, a color image contains red, blue, and green (RGB) channels, which are highly related to the image [18]. As a matter of fact, each pixel x of color image contains three gray pixels, i.e., $x = (x_r, x_g, x_b)$, where x_r, x_g, and x_b are RGB channels respectively. With a little changes of any channel, the color of x will have corresponding effects. The phenomenon of image degradation resulting from noise adversely affects the subsequent image processing and analysis, and visual effects [23, 26, 25]. Therefore, noise suppressing for improving color image quality is an essential process for many imaging tasks [37]. In this paper, we focus on the problem of removing additive Gaussian noise in color images. Mathematically, the degraded image $\mathbf{Y} \in \mathbb{R}^{m\times n}$ can be formulated as

$$\mathbf{Y} = \mathbf{X} + \mathbf{W}, \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the original image, and $\mathbf{W} \in \mathbb{R}^{m\times n}$ is the Gaussian white noise. In the past decades, many excellent denoising methods have been proposed, such as dictionary learning method [19], nonlocal means [3], block-matching and 3D filtering [9], and total variation [45, 46, 51], etc. We refer the reader to see [16] for a comprehensive review of the image denoising.

Received by the editors January 11, 2022.
2000 *Mathematics Subject Classification.* 68U10, 94A08, 90C47, 65K10.

Among the various denoising techniques, the dictionary-based method generalized K-means clustering for singular value decomposition (K-SVD) shows its superiority in reserving the textures, therefore, it has attracted considerable improvements in the last decade [11]. Indeed, Elad and Aharon [1] firstly proposed the effective patch-based method with K-SVD algorithm via sparse representation over a learned dictionary and updated the coefficients with orthogonal matching pursuit (OMP) algorithm. Given the noisy observation $\mathbf{Y}$, their model can be expressed as

$$\min_{\mathbf{D},\mathbf{a}_{ij},\mathbf{X}} \lambda\|\mathbf{X}-\mathbf{Y}\|_2^2+\sum_{i,j}(\mu_{ij}\|\mathbf{a}_{ij}\|_0+\|\mathbf{D}\mathbf{a}_{ij}-\mathcal{R}_{ij}\mathbf{X}\|_2^2), \tag{2}$$

where $\mathbf{D}\in\mathbb{R}^{m\times k}$ is the dictionary matrix, the $[i,j]$ indicates the image patch location, $\mathcal{R}_{ij}$ is an operator extracting the square $\sqrt{n}\times\sqrt{n}$ patch from the image at position $[i,j]$, and the vector $\mathbf{a}_{ij}\in\mathbb{R}^{k\times 1}$ is the coefficient vector for the corresponding patch with $\|\cdot\|_0$ being the ℓ_0-norm to count the nonzero number in the vector. As this method is designed for gray images initially, it will generate color distortion while be applied to color images by dealing with the three channels independently [50]. Hence, the patch-based dictionary method was improved to the patch group-based dictionary methods [48], which can eliminate the color bias. However, they still ignore the relationship among the color channels [50].

Recently, the quaternion representation has obtained much attention in image processing. The quaternion represents a color pixel by a structure, which can integrate the information of three channels. This advantage has promoted the application of quaternion representation in the color image processing [24]. For example, Yu et al. [54] applied quaternion-based weighted nuclear norm minimization (QWNNM) for color image denoising. The QWNNM model achieves better results than the real value-based weighted nuclear norm minimization method. Wang et al. [42] handled the color image segmentation with the quaternion-based method and has better results than the real value-based methods. Denoting a dot in variances as quaternion number and $\mathbb{H}$ as quaternion domain, the quaternion-based degradation model for color noise is given as

$$\dot{\mathbf{Y}}=\dot{\mathbf{X}}+\dot{\mathbf{W}}, \tag{3}$$

where $\dot{\mathbf{Y}}$, $\dot{\mathbf{X}}$, and $\dot{\mathbf{W}}\in\mathbb{H}^{m\times n}$ are the noisy image, latent clear image, and Gaussian white noise with zero mean and standard variance σ of quaternion form, respectively. The detailed information about quaternion please see Section 2.2. Comparing with vector-based models, the quaternion-based models fully utilize the relationship between channels and the orthogonal property for the coefficients of different channels [6] and thus generate better results. Due to the superiority of the quaternion-based method, Xu et al. [50] improved the model (2) with quaternion representation, and called it the K-QSVD model. Their idea is to fit color images with quaternion matrices and train the dictionary with the K-QSVD[1] and the QOMP[2] algorithms. Their K-QSVD model is formulated as follows

$$\min_{\dot{\mathbf{D}},\dot{\mathbf{a}}_{ij},\dot{\mathbf{X}}} \lambda\|\dot{\mathbf{X}}-\dot{\mathbf{Y}}\|_2^2+\sum_{i,j}(\mu_{ij}\|\dot{\mathbf{a}}_{ij}\|_0+\|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij}-\dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}\|_2^2), \tag{4}$$

where $\dot{\mathbf{D}}\in\mathbb{H}^{m\times k}$ is the dictionary matrix in quaternion form, the indicator $[i,j]$ marks the patch location, $\dot{\mathcal{R}}_{ij}$ is an operator extracting the square $\sqrt{n}\times\sqrt{n}$ patch

[1]The K-QSVD algorithm is the extension of the K-SVD algorithm, with all algebra operations in quaternion system.

[2]The QOMP algorithm is the extension of the OMP algorithm, with all algebra operations in quaternion system.

of coordinates $[i, j]$ from the image $\dot{\mathbf{X}}$, and the vector $\dot{\mathbf{a}}_{ij} \in \mathbb{H}^{k\times 1}$ is the coefficient vector for the corresponding patch. This patch-based dictionary learning method achieves better results with quaternion representation while there are still some limitations. The first comes from the calculation of quaternion numbers. Three imaginary parts and one real part compose a quaternion number. During the calculation of the quaternion-based algorithm, the real part will unavoidable be corrupted with some minor errors. This leads to the inappropriate representation of the color images. The second problem is the artifacts in images, especially when the noise level is high [12].

In this paper, we propose a novel approach to overcome the above-mentioned problems. Firstly, we propose and study an optimization model for color image denoising by enforcing the zero real part constraint in quaternion computation. A quaternion has four components (one real part and three imaginary parts), which increases the difficulty of calculation and brings a great challenge to establishing a pure quaternion-based dictionary learning model. Especially, the quaternion with four parts is a whole number. During the iteration, the real part of the quaternion will crop some unexpected numbers. Since the color image has three channels, we usually need to truncate the real part of the resulted quaternion matrix after the iteration, which leads to information loss. Different from [19], we investigate the pure quaternion-based sparse representation (pQS) method by adding a zero constraint to ensure that the color image is always represented as a pure quaternion matrix. In this case, the channel relationships and all the information of images can be well preserved at the same time. To overcome the second problem, we design an original quaternion-based total variation (q-TV) regularizer and study the denoising model based on pQS with q-TV regularizer, named by pQSTV model. This novel model can be solved by the alternating minimization method. dictionary learning part, due to the simplicity and efficiency of the K-SVD and OMP algorithms [1], we apply the quaternion-based K-SVD (K-QSVD) algorithm to learn the dictionary and the quaternion-based OMP (QOMP) algorithm to update coefficients.

The contribution of this paper is listed as follows:

- A new pure quaternion-based sparse representation (pQS) model is proposed for color image denoising, with a zero constraint on the real part. Without loss the geometric information of images, the structure of color channels is appropriately presented and preserved by this new model.
- A pure quaternion-based TV regularizer is firstly designed and embedded into the pQS model, which generates a pQSTV model. To the best of our knowledge, this pQSTV model is the first pure quaternion-based joint model to denoise the color image directly from the degraded image.
- Numerical results demonstrate clearly that the proposed model can provide better denoising results than the state-of-the-art methods, including K-QSVD, DnCNN, etc., by a large margin in average.

The outline of this paper is as follows. Section II recalls some basic concepts of quaternion algebra, dictionary learning, and the total variation method. Section III presents our approach. In Section IV, we display a series of experiments to compare the proposed method and other competitive methods. We conclude this work in Section V.

2. Related Works

2.1. Image recovery by dictionary. Various image processing methods have been proposed to denoise an image from its corrupted one. One popular class of denoising methods is based on dictionary learning and sparse coding, such as [10, 4]. If an image $\mathbf{X} \in \mathbb{R}^{m\times n}$ satisfies $\mathbf{X} = \mathbf{DA}$ (or $\mathbf{X} \approx \mathbf{DA}$), where $\mathbf{A} \in \mathbb{R}^{k\times n}$ is the sparse coefficient matrix (i.e., $\mathbf{A}$ has few nonzeros), then we call $\mathbf{X}$ is sparse (or approximately sparse) under a dictionary $\mathbf{D} \in \mathbb{R}^{m\times k}$ [13]. Many classes of images can be sparsely represented by different dictionaries [30]. Assuming that $\mathbf{X}$ is represented sparsely under a fixed dictionary $\mathbf{D}$, we can recover $\mathbf{X}$ via solving

$$\min_{\mathbf{A}} \|\mathbf{A}\|_0, \text{ s.t. } \|\mathbf{X} - \mathbf{DA}\|_2^2 \leq \epsilon, \tag{5}$$

where the ℓ_0-norm counts the number of non-zero elements and $\epsilon \geq 0$ is a parameter corresponding to the noise level. Once we get the solution of Eq. (5), i.e., the coefficient matrix $\mathbf{A}$, the ideal restored image $\mathbf{X}$ can be estimated by $\mathbf{DA}$. There are some predetermined dictionaries [41], such as overcomplete wavelets, discrete cosine transforms (DCT), and curvelets [49]. However, a learned dictionary can better represent the natural images and improve the recovery quality [13, 22]. A dictionary can be learned by algorithm K-SVD [1], MOD [14], and OLM [31], etc. With the character of simpleness and effectiveness [28], we train the dictionaries by the classical K-SVD method with the noisy image. The K-SVD method can be expressed by the following model

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{A}} \|\mathbf{X} - \mathbf{DA}\|_F^2, \\ &\text{s.t. } \|\mathbf{d}_i\|_2 = 1, i = 1, ..., k; \|\mathbf{a}_j\|_0 \leq \text{s}, j = 1, ...n, \end{aligned} \tag{6}$$

where $\mathbf{X} \in \mathbb{R}^{m\times n}$ represents the original sample, $\mathbf{d}_i$ is the i-th column of the trained dictionary $\mathbf{D}$. K-SVD method tries to solve Eq. (6) by alternatively updating $\mathbf{A}$ and $\mathbf{D}$ [1]. This problem can easily be solved by the Lasso (Least Absolute Shrinkage and Selection Operator) algorithm [40] and the OMP algorithm [35].

Aharon et al. [1] used the dictionary learning model to solve image denoising task, which generates better results than the predetermined dictionary. They reshape the color matrix as a large vector and treat an image as the linear connection of vectors, which ignores the correlation of image channels. Later, the quaternion matrix-based color image processing model is proposed in [50]. They represented color images with the quaternion matrix and completely preserved the inherent color structures during reconstruction. Next, we will review some concepts of the quaternion algebra and the quaternion's matrix and vector representation.

2.2. Quaternion algebra. A quaternion number [17] in quaternion domain $\mathbb{H}$ is expressed in the form

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

where a_0, a_1, a_2, and $a_3 \in \mathbb{R}$, $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ are the fundamental quaternion units which satisfy the quaternion rules

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1.$$

However, quaternion does not follow the multiplicatively commutative law, because $\mathbf{ij} = \mathbf{k}$, whereas $\mathbf{ji} = -\mathbf{k}$.

Let $\dot{a} = a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k} \in \mathbb{H}$, $\dot{b} = b_0 + b_1\mathbf{i} + b_2\mathbf{j} + b_3\mathbf{k} \in \mathbb{H}$, and $\lambda \in \mathbb{R}$, then we have

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

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

and

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

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

$$\dot{a}^* = a_0 - a_1\mathbf{i} - a_2\mathbf{j} - a_3\mathbf{k},$$
$$|\dot{a}| = \sqrt{a_0^2 + a_1^2 + a_2^2 + a_3^2}.$$

The quaternion matrix is a matrix whose entries are elements of the quaternion's algebra. Suppose $\dot{\mathbf{Q}}$ is a quaternion matrix, i.e., $\dot{\mathbf{Q}} \in \mathbb{H}^{m\times n}$, then

$$\dot{\mathbf{Q}} = \mathbf{Q}_0 + \mathbf{Q}_1\mathbf{i} + \mathbf{Q}_2\mathbf{j} + \mathbf{Q}_3\mathbf{k}, \tag{7}$$

where $\mathbf{Q}_0$, $\mathbf{Q}_1$, $\mathbf{Q}_2$, and $\mathbf{Q}_3 \in \mathbb{R}^{m\times n}$. Therefore, the RGB channels of a color pixel $\dot{\mathbf{q}}_{ij}$ can be encoded as the three imaginary parts of the quaternion [36]

$$\dot{\mathbf{q}}_{ij} = \mathbf{r}_{ij}\mathbf{i} + \mathbf{g}_{ij}\mathbf{j} + \mathbf{b}_{ij}\mathbf{k}, \tag{8}$$

where $i = 1, \ldots, m$, $j = 1, \ldots, n$, $\dot{\mathbf{q}}_{ij} \in \mathbb{H}$ is a pure quaternion number (i.e., without real component), and $\mathbf{r}_{ij}$, $\mathbf{g}_{ij}$, and $\mathbf{b}_{ij}$ are the RGB channels corresponding to a pixel in the color image.

The norms of quaternion matrix and vector are defined as follows.

Definition 2.1. The ℓ_2-norm of quaternion vector $\dot{\mathbf{a}} = \alpha_0 + \alpha_1\mathbf{i} + \alpha_2\mathbf{j} + \alpha_3\mathbf{k} \in \mathbb{H}^n$ is $\|\dot{\mathbf{a}}\|_2 := \sqrt{\sum_i |\alpha_i|^2}$; the ℓ_2-norm of quaternion matrix $\dot{\mathbf{Q}} = (\dot{\mathbf{q}}_{ij})_{m\times n}$ is $\|\dot{\mathbf{Q}}\|_2 := \max(\dot{\sigma}_i)$, where $\dot{\sigma}_i$ is the set of singular values of $\dot{\mathbf{Q}}$, $i = 1, ..., s$, and the Frobenius norm is $\|\dot{\mathbf{Q}}\|_{\mathrm{F}} := \sqrt{\sum_{i,j} |\dot{\mathbf{q}}_{ij}|^2}$.

The singular value decomposition (SVD) of a quaternion matrix was firstly proposed in [55].

Theorem 2.2. ***(Quaternion Singular Value Decomposition (QSVD))*** *Let $\dot{\mathbf{Q}} \in \mathbb{H}^{m\times n}$, then there exist two unitary quaternion matrices $\dot{\mathbf{U}} \in \mathbb{H}^{m\times m}$ and $\dot{\mathbf{V}} \in \mathbb{H}^{n\times n}$ such that $\dot{\mathbf{U}}^*\dot{\mathbf{Q}}\dot{\mathbf{V}} = \dot{\Sigma}$, where $\dot{\Sigma} = diag(\dot{\sigma}_1, \dot{\sigma}_2, \ldots, \dot{\sigma}_{\mathrm{s}})$, with $|\dot{\sigma}_i| \geq 0$ and $\mathrm{s} = min(\mathrm{m}, \mathrm{n})$.*

Based on the above definition, the quaternion-based model (4) can be well handled. In [50], Xu et al. proposed the QOMP and the K-QSVD algorithm to solve their quaternion-based model and reported competitive results by representing images with the quaternion matrix in color image processing. The dictionary learning prior in the quaternion domain can match similar patches information in color images and generate promising denoising results [50]. In [1], the authors apply the OMP and the K-SVD algorithms to solve the dictionary learning-based model. However, their results generate unexpected artifacts. Considering the effectiveness of TV in suppressing artifacts [7], we consider combining the dictionary learning method and the TV regularizer for color image denoising. Next, we will give a brief introduction to the TV term.

2.3. Quaternion-based total variation regularizer. The total variation [38] was designed for grayscale image processing and has become one of the most popular regularization methods in grayscale image processing. In the last decades, total variation has been developed to many other forms for image processing problems [29]. For example, we have high-order TV [21], weighted TV [8], anisotropic TV

[34], and nonlocal TV [24], etc. The TV model proposed in [38] can be expressed as

$$\mathbf{X} = \arg\min_{\mathbf{X}} \eta \mathcal{J}_{TV}(\mathbf{X}) + \frac{1}{2}\|\mathbf{Y} - \mathbf{X}\|_2^2, \tag{9}$$

where $\frac{1}{2}\|\mathbf{Y} - \mathbf{X}\|_2^2$ is the fidelity term and $\mathcal{J}_{TV}(\mathbf{X})$ is the regularization term, η is the regular parameter.

There are two popular types of regularization terms. One is the ℓ_2-based isotropic TV [38] defined as

$$\mathcal{J}_{TV}(\mathbf{X}) = \|\nabla \mathbf{X}\|_2 = \sqrt{\mathbf{X}_s^2 + \mathbf{X}_t^2}, \tag{10}$$

and the other is the ℓ_1-based anisotropic TV [33] defined as

$$\mathcal{J}_{TV}(\mathbf{X}) = \|\nabla \mathbf{X}\|_1 = |\mathbf{X}_s| + |\mathbf{X}_t|, \tag{11}$$

where $\nabla = (\frac{\partial}{\partial s}, \frac{\partial}{\partial t})$ is the gradient operator and $\nabla \mathbf{X} = (\mathbf{X}_s, \mathbf{X}_t)$. Here, $\mathbf{X}_s$ and $\mathbf{X}_t$ are the gradients of $\mathbf{X}$ in the directions of s and t, respectively. And $\mathcal{J}_{TV}(\mathbf{X})$ denotes the total variation of $\mathbf{X}$. When it comes to the quaternion domain, the ℓ_1 and ℓ_2-based TV of a quaternion matrix $\dot{\mathbf{X}} = \mathbf{X}_0 + \mathbf{X}_1\mathbf{i} + \mathbf{X}_2\mathbf{j} + \mathbf{X}_3\mathbf{k} \in \mathbb{H}^{m\times n}$ can be defined as

$$\|\nabla \dot{\mathbf{X}}\|_1 := \|\nabla \mathbf{X}_0\|_1 + \|\nabla \mathbf{X}_1\|_1 + \|\nabla \mathbf{X}_2\|_1 + \|\nabla \mathbf{X}_3\|_1, \tag{12}$$

and

$$\|\nabla \dot{\mathbf{X}}\|_2 := \|\nabla \mathbf{X}_0\|_2 + \|\nabla \mathbf{X}_1\|_2 + \|\nabla \mathbf{X}_2\|_2 + \|\nabla \mathbf{X}_3\|_2. \tag{13}$$

By the number of numerical experiments, we find that the ℓ_1-based TV regularization overcomes the grid artifacts well and it also costs less computational flops than the ℓ_2-based TV regularization. So that we concentrate on the definition (12) and denote it by the q-TV in this paper.

3. Pure quaternion-based sparse representation TV model

In this section, we present a novel color image denoising model based on pure quaternion-based sparse representation and TV regularization.

3.1. Zero constraint $\mathrm{Re}(\dot{\mathbf{X}}) = 0$.

Let quaternion matrix $\dot{\mathbf{X}} = \mathbf{X}_0 + \mathbf{X}_1\mathbf{i} + \mathbf{X}_2\mathbf{j} + \mathbf{X}_3\mathbf{k} \in \mathbb{H}^{m\times n}$ represent a color image, where the real part $\mathbf{X}_0 \in \mathbb{R}^{m\times n}$ is zero, and three imaginary parts $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{X}_3 \in \mathbb{R}^{m\times n}$ denote the red, green, and blue channels, respectively. Due to the errors of truncation and rounding, the quaternion matrices generated by the proposed algorithm have absolutely small nonzero entries in their real parts. In traditional methods, the nonzero real part is often cut off at the outputting step, which makes the reconstructed color images have sight color distortion. To solve this deficiency, the solution of our model will be constrained to be a pure quaternion matrix representing a color image with three color channels. That is, all quaternion matrices generated in the solving process will be forced to have zero real parts. Such zero constraint is described by $\mathrm{Re}(\dot{\mathbf{X}}) = 0$. To simplify the computation, an indicator function Φ_0 is introduced to the set of pure quaternion matrices, $\{\dot{\mathbf{X}} \mid \mathbf{X}_0 = 0, \dot{\mathbf{X}} = \mathbf{X}_0 + \mathbf{X}_1\mathbf{i} + \mathbf{X}_2\mathbf{j} + \mathbf{X}_3\mathbf{k}\}$. The zero constraint can be reformulated as $\Phi_0(\dot{\mathbf{X}}) = 0$. Explicitly, let $\dot{\mathbf{X}} = \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix}$, then

$$\Phi_0(\dot{\mathbf{X}}) = \Phi_0 \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix} = \mathbf{X}_0.$$

3.2. The pQS Representation Model and Algorithm. Now, we study the image denoising problem under Gaussian noise. Assuming that the degraded image $\dot{\mathbf{Y}} \in \mathbb{H}^{m\times n}$ is formulated through Eq. (3), we propose the following pure quaternion-based sparse representation (pQS) model

$$\begin{aligned} &\min_{\dot{\mathbf{D}},\dot{\mathbf{a}}_{ij},\dot{\mathbf{X}}} \lambda\|\dot{\mathbf{X}} - \dot{\mathbf{Y}}\|_2^2 + \sum_{i,j} \mu_{ij}\|\dot{\mathbf{a}}_{ij}\|_0 + \sum_{i,j} \|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij} - \dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}\|_2^2, \\ &\text{s.t. } \mathrm{Re}(\dot{\mathbf{X}}) = 0, \end{aligned} \tag{14}$$

where again $\dot{\mathbf{D}} \in \mathbb{H}^{m\times k}$ is the dictionary matrix, the indicators $[i, j]$ mark the location of the patches in the image, $\dot{\mathcal{R}}_{ij}$ is an extracting operator, and the vectors $\dot{\mathbf{a}}_{ij} \in \mathbb{H}^{k\times 1}$ are the coefficient vectors for the corresponding patches. $\|\cdot\|_0$ is the ℓ_0 norm. As far as our knowledge goes, the real part is firstly constrained to be zero in the above dictionary learning model to ensure that the color image is represented as a pure quaternion matrix.

One of the advantages of constraining the real part of the quaternion matrix to be zero is that one can fit color images perfectly. Even a tiny real part will cause a loss of color information. Indeed, restricting the real part to zero has more improvements than the quaternion matrix method. In Fig. 1, we display the two dictionaries trained by the K-QSVD algorithm (i.e., [50]) and the proposed pure quaternion-based dictionary learning model, respectively. From Fig. 1, we can see that the dictionary on the left is not rich enough to denote the color image perfectly, which leads to having artifacts in the restored image. In contrast, the trained dictionary on the right is instead approaching the color image, which preserves the connection of RGB channels and shows the image's faultlessness.

The details of solving the Eq. (14) are as follows.

- We first give the dictionary $\dot{\mathbf{D}}$, the coefficient of every image patch is

$$\min_{\dot{\mathbf{a}}} \mu_{ij}\|\dot{\mathbf{a}}_{ij}\|_0 + \|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij} - \dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}_{ij}\|_2^2. \tag{15}$$

  The QOMP algorithm can deal with this problem.
- Given the initial image $\dot{\mathbf{X}}$, then we have

$$\min_{\dot{\mathbf{D}}} \sum_{i,j} \mu_{ij}\|\dot{\mathbf{a}}_{ij}\|_0 + \sum_{i,j} \|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij} - \dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}\|_2^2. \tag{16}$$

  The K-QSVD algorithm can handle this problem by updating the dictionary $\dot{\mathbf{D}}$ and $\dot{\mathbf{a}}$, alternatively.
- Given the dictionary $\dot{\mathbf{D}}$ and all coefficient $\dot{\mathbf{a}}_{ij}$, we can update $\dot{\mathbf{X}}$ by

$$\begin{aligned} &\min_{\dot{\mathbf{X}}} \lambda\|\dot{\mathbf{X}} - \dot{\mathbf{Y}}\|_2^2 + \sum_{ij} \|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij} - \dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}\|_2^2, \\ &\text{s.t. } \mathrm{Re}(\dot{\mathbf{X}}) = 0. \end{aligned} \tag{17}$$

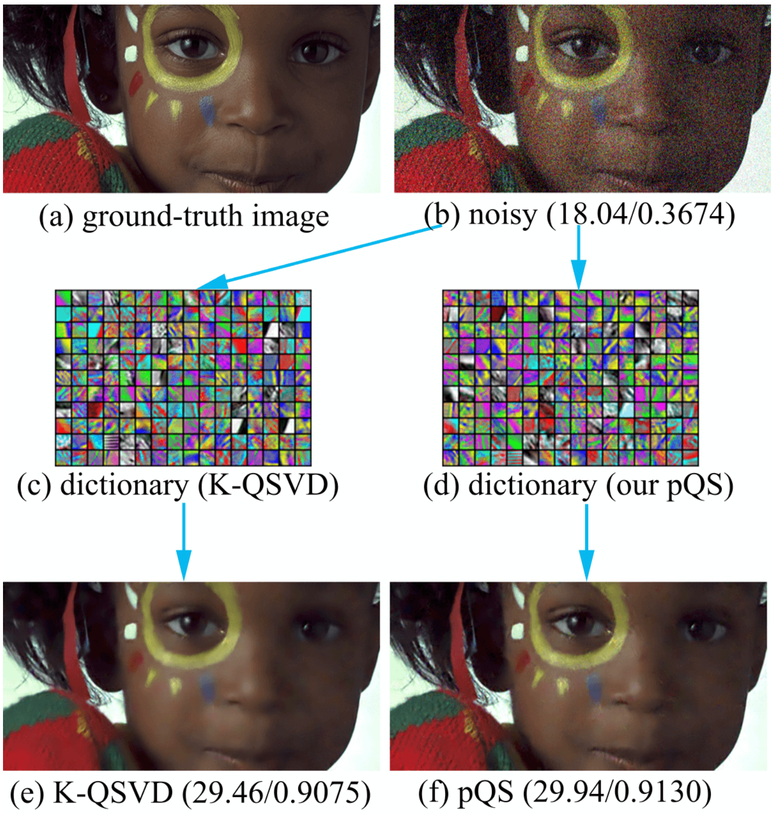


FIGURE 1. Display of different dictionaries, the left column is the process of dictionary trained by the K-SVD algorithm with a quaternion matrix, and the right column is with a pure quaternion matrix. The noisy image is covered with Gaussian noise and the noise level is 35.

We apply the alternating minimization method (AMM) to solve Eq. (17). At first, the constrained optimization problem (17) is derived into the following unconstrained optimization problem

$$\min_{\dot{\mathbf{X}}} \lambda\|\dot{\mathbf{X}}-\dot{\mathbf{Y}}\|_2^2+\sum_{ij}\|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij}-\dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}\|_2^2+\xi\|\Phi_0(\dot{\mathbf{X}})\|_2^2, \tag{18}$$

where $\sum_{ij}\|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij}-\dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}\|_2^2$ is differentiable (see [50]), and ξ is a positive parameter. Let

$$\dot{\mathbf{Y}}=\begin{bmatrix}\mathbf{Y}_0\\ \mathbf{Y}_1\\ \mathbf{Y}_2\\ \mathbf{Y}_3\end{bmatrix}, \dot{\mathbf{D}}=\begin{bmatrix}\mathbf{D}_0\\ \mathbf{D}_1\\ \mathbf{D}_2\\ \mathbf{D}_3\end{bmatrix}, \dot{\mathbf{a}}_{ij}=\begin{bmatrix}\mathbf{a}_{0_{ij}}\\ \mathbf{a}_{1_{ij}}\\ \mathbf{a}_{2_{ij}}\\ \mathbf{a}_{3_{ij}}\end{bmatrix}, \dot{\mathcal{R}}_{ij}=\begin{bmatrix}\mathcal{R}_{0_{ij}}\\ \mathcal{R}_{1_{ij}}\\ \mathcal{R}_{2_{ij}}\\ \mathcal{R}_{3_{ij}}\end{bmatrix}, \tag{19}$$

then we reformulate the Eq. (18) as

$$\min_{\mathbf{X}_0,\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3} \lambda \left\| \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix} - \begin{bmatrix} \mathbf{Y}_0 \\ \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \mathbf{Y}_3 \end{bmatrix} \right\|_2^2 + \xi \left\| \Phi_0 \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix} \right\|_2^2 + \sum_{ij} \left\| \begin{bmatrix} \mathbf{D}_0 \\ \mathbf{D}_1 \\ \mathbf{D}_2 \\ \mathbf{D}_3 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{a}_{0_{ij}} \\ \mathbf{a}_{1_{ij}} \\ \mathbf{a}_{2_{ij}} \\ \mathbf{a}_{3_{ij}} \end{bmatrix} - \begin{bmatrix} \mathcal{R}_{0_{ij}} \\ \mathcal{R}_{1_{ij}} \\ \mathcal{R}_{2_{ij}} \\ \mathcal{R}_{3_{ij}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix} \right\|_2^2 . \tag{20}$$

Note that

$$\Phi_0(\dot{\mathbf{X}}) = \Phi_0 \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix} = \mathbf{X}_0,$$

then we have

$$\min_{\mathbf{X}_0,\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3} \lambda \sum_{\iota=0}^{3} \|\mathbf{X}_\iota - \mathbf{Y}_\iota\|_2^2 + \xi \|\mathbf{X}_0\|_2^2 + \sum_{ij} \sum_{\iota=0}^{3} \|\mathbf{D}_\iota \mathbf{a}_{\iota_{ij}} - \mathcal{R}_{\iota_{ij}} \mathbf{X}_\iota\|_2^2 . \tag{21}$$

The above minimization problem (21) has a closed-form solution

$$\dot{\mathbf{X}} = \frac{\lambda \dot{\mathbf{Y}} + \sum_{ij} \dot{\mathcal{R}}_{ij}^* \dot{\mathbf{D}} \dot{\mathbf{a}}_{ij}}{\Xi \dot{\mathbf{I}} + \sum_{ij} \dot{\mathcal{R}}_{ij}^* \dot{\mathcal{R}}_{ij}}, \tag{22}$$

where $\Xi = (\xi + \lambda, \lambda, \lambda, \lambda)^* \in \mathbb{H}^{4\frac{m}{4} \times 1 \cdot n}$ is a column vector.

The process of our pQS method is shown in Algorithm 1.

Algorithm 1 Color image denoising algorithm with our pQS model

---

Require:

The noisy image $\mathbf{Y} \in \mathbb{R}^{m \times n}$;

Parameter $\lambda \in \mathbb{R}$, iteration numbers N and M;

Ensure:

The denoised image $\mathbf{X} \in \mathbb{R}^{m \times n}$;

```
Initialization: Representing Y as quaternion matrix Ẏ ∈ H^{m×n}. Randomly
choose column vector {ḋ_1, ḋ_2, ..., ḋ_k} from Ẏ as the initial dictionary Ḋ^(0). Let
the coefficient vectors ȧ_ij = 0;
for t = 1 : N  do
  Calculate ȧ^t_ij by Eq. (15);
  Update Ḋ^t and ȧ^t_ij by Eq. (16);
  for k = 1 : M do
    Update Ẋ^{k+1} by Eq. (22);
    k = k + 1;
  end for
  t = t + 1;
end for
return  Ẋ
```

3.3. The pQS Representation Model with q-TV Regularization and Algorithm. In the strong noising case, the direct patch-based method may yield artifacts. For instance, we test the color images with a high Gaussian noise level (σ=50). Fig. 2 displays the restored results of our pQS method (the fourth column) and the K-QSVD method (the third column). We can see that constraining the real part of the quaternion matrix to be zero indeed works, and the improvement can be seen from the visual quality and numerical results. Unfortunately, the restored image that Eq. (14) recovered still has some grid artifacts, especially for images degraded by high-level noise.

We need careful treatment for reducing these artifacts. Indeed, the sparse representation is good at preserving texture while the TV method can smooth artifacts at the cost of slightly affecting the texture information. Hence, it is interesting to check whether the combination of TV and sparse representation can improve the restoration results. This can be regarded as one of our main contributions. Indeed, in order to give a better evaluation of the pQS method, we can further improve the model (14) by introducing the quaternion-TV regularization term. In Fig. 2, we find that joint sparse representation and the q-TV regularizer have good results in color image denoising. By designing a quaternion based TV as $\|\nabla\dot{\mathbf{X}}\|_1 = \|\nabla\mathbf{X}_0\|_1 + \|\nabla\mathbf{X}_1\|_1 + \|\nabla\mathbf{X}_2\|_1 + \|\nabla\mathbf{X}_3\|_1$, the proposed pQSTV model can be written as

$$\begin{aligned} &\min_{\dot{\mathbf{D}},\dot{\mathbf{a}}_{ij},\dot{\mathbf{X}}} \lambda\|\dot{\mathbf{X}}-\dot{\mathbf{Y}}\|_2^2 + \eta\|\nabla\dot{\mathbf{X}}\|_1 + \sum_{i,j}(\mu_{ij}\|\dot{\mathbf{a}}_{ij}\|_0 + \|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij} - \dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}\|_2^2),\\ &\qquad \text{s.t. } \mathrm{Re}(\dot{\mathbf{X}}) = 0, \end{aligned} \tag{23}$$

where $\eta \in \mathbb{R}$ is the regularization parameter, $\|\nabla\dot{\mathrm{X}}\|_1$ is the quaternion-based TV regularization. Since the ℓ_1-based TV regularization overcomes the grid artifacts already, we will not discuss ℓ_2-based TV regularization here. With the help of TV regularization, Eq. (23) can stabilize the recovered results. The visual quality and numerical results are shown in Fig. 2. Clearly, the recovered images show that the artifacts are eliminated completely.

The remaining problem is how to efficiently solve the optimization problem (23). Actually, it is not easy to solve Eq. (23), since the q-TV regularization is not differentiable and this model is nonconvex. Fortunately, there are many methods to solve this problem in these days. Here, we try to solve Eq. (23) by using the variable splitting method [52]. Using the alternating minimization method and quaternion rules, we try to solve the proposed model. The subproblems are listed as follows.

- Given $\dot{\mathbf{D}}, \dot{\mathbf{X}}$, the minimization for $\dot{\mathbf{a}}_{ij}$ satisfies

$$\min_{\dot{\mathbf{a}}_{ij}} \mu_{ij}\|\dot{\mathbf{a}}_{ij}\|_0 + \|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij} - \dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}\|_2^2. \tag{24}$$

  We can use the QOMP method to deal with the above subproblem.
- Given $\dot{\mathbf{X}}, \dot{\mathbf{a}}_{ij}$, the minimization for $\dot{\mathbf{D}}$ satisfies

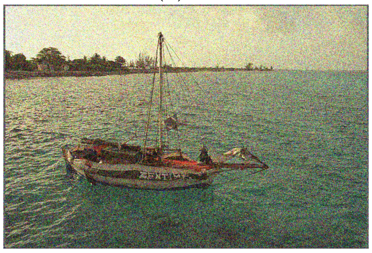
$$\min_{\dot{\mathbf{D}}} \sum_{i,j} \|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij} - \dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}\|_2^2. \tag{25}$$

  The K-QSVD method can effectively address above minimization which stops searching the best candidate atom when approximation reaches the sphere of radius $\sqrt{\epsilon}$ in Eq. (5).
- Given $\dot{\mathbf{Y}}, \dot{\mathbf{a}}_{ij}$ and $\dot{\mathbf{D}}$, the minimization for $\dot{\mathbf{X}}$ satisfies

$$\begin{aligned} &\min_{\dot{\mathbf{X}}} \lambda\|\dot{\mathbf{X}}-\dot{\mathbf{Y}}\|_2^2 + \eta\|\nabla\dot{\mathbf{X}}\|_1 + \sum_{ij}\|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij} - \dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}\|_2^2,\\ &\qquad \text{s.t. } \mathrm{Re}(\dot{\mathbf{X}}) = 0. \end{aligned} \tag{26}$$

(a) K6

(a) K3

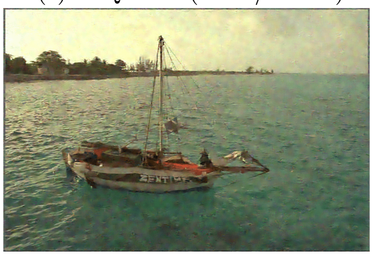


(b) noisy (14.15/0.2792)

(b) noisy (14.15/0.2595)

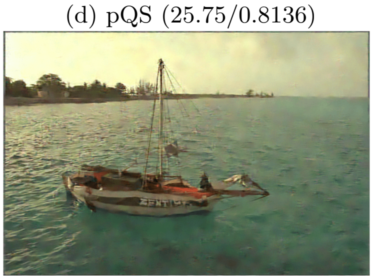
(c) K-QSVD$^+$(25.36/0.7962)

(c) K-QSVD$^+$(28.97/0.9184)

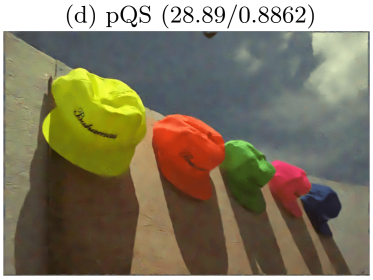


(d) pQS (25.75/0.8136)

(d) pQS (28.89/0.8862)



(e) pQSTV (25.92/0.8232)



(e) pQSTV (29.81/0.9310)

FIGURE 2. Color image denoising results on K3 and K6 with P-SNR/SSIM. (a) Original image; (b) Noisy image corrupted by Gaussian noise with variance $\sigma = 50$; The denoised image reconstructed by: (c) K-QSVD$^+$(Eq.(4)), (d) Our pQS (Eq. (14)), (e) pQSTV (Eq. (23)).

We give the existence and uniqueness of the solution of problem (26) in Proposition 1. According to Eq. (19), we rewrite (26) into

$$\min_{\mathbf{X}_0,\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3} \lambda\sum_{\iota=0}^{3}\|\mathbf{X}_\iota-\mathbf{Y}_\iota\|_2^2+\eta\sum_{\iota=0}^{3}\|\nabla_\iota\mathbf{X}_\iota\|_1+\sum_{ij}\sum_{\iota=0}^{3}\|\mathbf{D}_\iota\mathbf{a}_{\iota_{ij}}-\mathcal{R}_{\iota_{ij}}\mathbf{X}_\iota\|_2^2, \quad \text{s.t. } \mathrm{Re}(\dot{\mathbf{X}})=0. \tag{27}$$

We reformulate Eq. (27) to an unconstrained problem as

$$\begin{aligned}\min_{\mathbf{X}_0,\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3} &\lambda\sum_{\iota=0}^{3}\|\mathbf{X}_\iota-\mathbf{Y}_\iota\|_2^2+\eta\sum_{\iota=0}^{3}\|\nabla_\iota\mathbf{X}_\iota\|_1\\ &+\sum_{ij}\sum_{\iota=0}^{3}\|\mathbf{D}_\iota\mathbf{a}_{\iota_{ij}}-\mathcal{R}_{\iota_{ij}}\mathbf{X}_\iota\|_2^2+\xi_1\|\Phi_0(\dot{\mathbf{X}})\|_2^2,\end{aligned} \tag{28}$$

where Φ_0 denotes the indicator function of the set of pure quaternion matrices. Denote the subjection of the minimization (28) by $J(\mathbf{X})$. Recall the definition of coerciveness as follows.

Definition 3.1. A function $J : T \to \mathbb{R}$ on a Banach space T is called coercive if $\|\mathbf{X}^k\| \to +\infty$ implies $J(\mathbf{X}^k) \to +\infty$ for every sequence $\{\mathbf{X}^k\}_{k\in\mathbb{N}} \subset T$.

The existence of a solution to the proposed model (28) is based on the theorem that any continuous, convex, and coercive function on a Banach space has a global minimizer [2].

Proposition 1. *There exists a unique minimizer for the objective function in (28).*

The proof of Proposition 1 can be found in the supplementary material Appendix.

The AMM algorithm can handle Eq. (26). By introducing the auxiliary quaternion variable $\dot{\mathbf{p}}$, then the above minimization problem can be reformulated as

$$\min_{\dot{\mathbf{X}},\dot{\mathbf{p}}} \lambda\|\dot{\mathbf{X}}-\dot{\mathbf{Y}}\|_2^2+\eta\|\dot{\mathbf{p}}\|_1+\frac{\eta_1}{2}\|\nabla\dot{\mathbf{X}}-\dot{\mathbf{p}}\|_2^2+\sum_{ij}\|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij}-\dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}\|_2^2+\xi_1\|\mathrm{Re}(\dot{\mathbf{X}})\|_2^2, \tag{29}$$

where λ, η, η_1, ξ_1 are positive parameters. For fixed $\dot{\mathbf{X}}$, the minimization for $\dot{\mathbf{p}}$ is an L_1-regularized least square problem

$$\min_{\dot{\mathbf{p}}} \eta\|\dot{\mathbf{p}}\|_1+\frac{\eta_1}{2}\|\nabla\dot{\mathbf{X}}-\dot{\mathbf{p}}\|_2^2. \tag{30}$$

It can be solved by the least absolute shrinkage (see [43])

$$\dot{\mathbf{p}} = \mathrm{shrink}(\nabla\dot{\mathbf{X}}, \frac{\eta}{\eta_1}), \tag{31}$$

and the shrinkage operator can be defined as

$$\mathrm{shrink}(x,\tau)_{ij} := \max(\|x_{ij}\|_2-\tau,0)\frac{x_{ij}}{\|x_{ij}\|_2}, \tag{32}$$

with x_{ij} denoting the ij-th component of x. Given $\dot{\mathbf{p}}$, the subproblem for $\dot{\mathbf{X}}$ is a least squares problem

$$\min_{\dot{\mathbf{X}}} \lambda\|\dot{\mathbf{X}}-\dot{\mathbf{Y}}\|_2^2+\frac{\eta_1}{2}\|\nabla\dot{\mathbf{X}}-\dot{\mathbf{p}}\|_2^2+\sum_{i,j}\|\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij}-\dot{\mathcal{R}}_{ij}\dot{\mathbf{X}}\|_2^2+\xi_1\|\Phi_0(\dot{\mathbf{X}})\|_2^2. \tag{33}$$

According to Eq. (21), the above minimization problem can be written as

$$\begin{aligned}\min_{\mathbf{X}_0,\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3} &\lambda\sum_{\iota=0}^{3}\|\mathbf{X}_\iota-\mathbf{Y}_\iota\|_2^2+\frac{\eta_1}{2}\sum_{\iota=0}^{3}\|\nabla\mathbf{X}_\iota-\mathbf{p}_\iota\|_2^2\\ &+\sum_{i,j}\sum_{\iota=0}^{3}\|\mathbf{D}_\iota\mathbf{a}_{\iota_{ij}}-\mathcal{R}_{\iota_{ij}}\mathbf{X}_\iota\|_2^2+\xi_1\|\mathbf{X}_0\|_2^2.\end{aligned} \tag{34}$$

Then we have

$$2\lambda(\dot{\mathbf{X}}-\dot{\mathbf{Y}})+\eta_1\nabla^*(\nabla\dot{\mathbf{X}}-\dot{\mathbf{p}})+2\sum_{ij}\mathcal{R}_{ij}^*(\mathcal{R}_{ij}\dot{\mathbf{X}}-\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij})+2\xi_1\mathbf{X}_0=0. \tag{35}$$

The closed-form solution is as follows

$$\dot{\mathbf{X}}=\frac{2\lambda\dot{\mathbf{Y}}+\eta_1\nabla^*\dot{\mathbf{p}}+2\sum_{ij}\mathcal{R}_{ij}^*\dot{\mathbf{D}}\dot{\mathbf{a}}_{ij}}{\Xi\dot{\mathbf{I}}+\eta_1\nabla^*\nabla+2\sum_{ij}\mathcal{R}_{ij}^*\mathcal{R}_{ij}}, \tag{36}$$

where $\Xi=(2\lambda+2\xi_1,2\lambda,2\lambda,2\lambda)^*\in\mathbb{H}^{4\frac{m}{4}\times 1\cdot n}$ is a column vector.

The whole procedure of our pQSTV (pure Quaternion Sparse Total Variation) method is shown in Algorithm 2.

Algorithm 2 Color image denoising algorithm with our pQSTV model (23)

Require:
The noisy image $\mathbf{Y}\in\mathbb{R}^{m\times n}$;
Parameters $\lambda,\eta,\eta_1\in\mathbb{R}$, iteration numbers N and M;
Ensure:
The Denoised image $\mathbf{X}$;
Initialization: Representing $\mathbf{Y}$ as quaternion matrix $\dot{\mathbf{Y}}\in\mathbb{H}^{m\times n}$. Randomly choose column vector $\{\dot{\mathbf{d}}_1,\dot{\mathbf{d}}_2,...\dot{\mathbf{d}}_k\}$ from $\dot{\mathbf{Y}}$ as the initial dictionary $\dot{\mathbf{D}}^{(0)}$. Let the coefficient vectors $\dot{\mathbf{a}}_{ij}=0$;
for $\mathrm{t}=1:\mathrm{N}$ **do**
  Update $\dot{\mathbf{a}}_{i,j}^{\mathrm{t}}$ by equation (24);
  Update $\dot{\mathbf{D}}^{\mathrm{t}}$ by equation (25);
  for $\mathrm{k}=1:\mathrm{M}$ **do**
    Update $\dot{\mathbf{p}}^{\mathrm{k}+1}$ by equation (31);
    Update $\dot{\mathbf{X}}^{\mathrm{k}+1}$ by equation (36);
    $\mathrm{k}=\mathrm{k}+1$;
  end for
  $\mathrm{t}=\mathrm{t}+1$;
end for
return $\dot{\mathbf{X}}$

FIGURE 3. Images in Kodak24 database.

4. Experiments

In this section, we first illustrate the experimental details and then compare the proposed pQS and pQSTV methods to other state-of-the-art color image denoising methods including ℓ_1-ROF [5], SV-TV [20], CEM [27], the improved K-SVD denoising method [32], K-QSVD method [50], K-QSVD$^+$[50], PGPD [48], and DnCNN [56]. The comparisons are conducted on three datasets, i.e., Kodak Image Dataset[3] with image size 768×512 and 512×768, Set5[4] with image size S1 (512×512), S2 (256×256), S3 (280×280), S4 (288×288), S5 (228×344), and CSet8[5] with image size 256×256, which are shown in Fig. 3, Fig. 4 and Fig. 5, respectively. In our experiments, the noisy images are synthesized with Eq. (3) by adding the additive white Gaussian noise with variance σ to the clean color images. For the numerical comparison, we use the structural similarity index (SSIM) and the peak signal-to-noise ratio (PSNR) [44] to measure the quality of the restored images from the noisy images by different methods. Note that all the simulations are run in Matlab R2020a on a 64-bit workstation with a 3.70GHz CPU and 8GB memory.

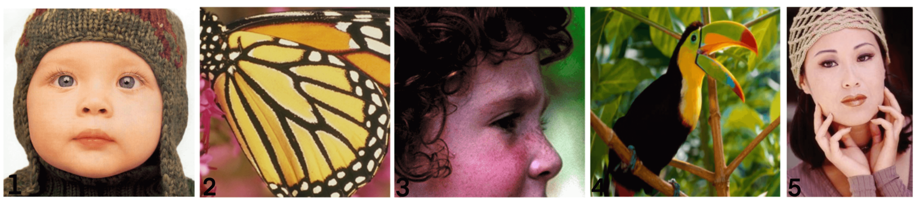


FIGURE 4. Images in Set5 database.

[3]http://www.r0k.us/graphics/kodak/

[4]http://people.rennes.inria.fr/Aline.Roumy/results/SR_BMVC12.html

[5]https://github.com/ysix7/Dataset

FIGURE 5. Images in CSet8 database.

4.1. Parameter setting. For the patch-based methods K-QSVD and proposed pQSTV, we firstly adjust the overlapping patch size from 8×8 to 12×12 for each image from Kodak24 in different noise levels to find out the one for the best-restored results. The best results among different patch sizes for pQSTV and K-QSVD are denoted by '$\star$' in Table 1 and Table 2, respectively. As one can see, for the proposed model pQSTV, the patch size 9×9 performs best for most images in the case of lower noise levels while the patch size 11×11 is better in the case of the higher noise level. For the method K-QSVD, we select the patch size 9×9 for noise 25, 10×10 for noise 35, and 8×8 for noise 50. According to this observation, in the following Table 3, 4, and 5, we apply these selected patch sizes to perform the proposed pQS, pQSTV and K-QSVD. Note that we denote the method K-QSVD with the selected patch size by K-QSVD$^+$ while K-QSVD denotes the method we test with the default parameter defined in [50].

TABLE 1. Distribution of patchsize in our pQSTV (Kodak24).

Patchsize \ Images, σ		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
$\sigma = 25$	8×8					$\star$	$\star$							$\star$											$\star$
	9×9	$\star$							$\star$	$\star$	$\star$	$\star$			$\star$	$\star$			$\star$			$\star$	$\star$		
	10×10							$\star$					$\star$				$\star$	$\star$							
	11×11		$\star$	$\star$	$\star$															$\star$	$\star$			$\star$	
	12×12																								
$\sigma = 35$	8×8				$\star$									$\star$					$\star$						
	9×9					$\star$	$\star$	$\star$				$\star$			$\star$							$\star$			$\star$
	10×10																	$\star$		$\star$					
	11×11	$\star$	$\star$	$\star$					$\star$	$\star$	$\star$		$\star$			$\star$	$\star$				$\star$		$\star$	$\star$	
	12×12																								
$\sigma = 50$	8×8																								$\star$
	9×9					$\star$								$\star$	$\star$				$\star$			$\star$			
	10×10															$\star$		$\star$							
	11×11	$\star$	$\star$		$\star$			$\star$	$\star$	$\star$	$\star$	$\star$					$\star$			$\star$	$\star$		$\star$	$\star$	
	12×12			$\star$			$\star$						$\star$												

Apart from the patch size, the iteration number N of K-QSVD, K-QSVD$^+$, and our models (pQS and pQSTV) is set to be 1. For the proposed pQS and pQSTV, the inner iteration number M of the proposed pQS and pQSTV is set to be 1 with other parameters $\lambda = 0.037$, $\eta = 0.1$, $\eta_1 = 0.5$, $\xi = \xi_1 = 1$. All the parameters of other competing methods are set as the default values as given in their codes and papers. As to the data-driven method DnCNN [56], the training part of the DnCNN was not considered in this paper, we apply the Matlab function 'denoisingNetwork('Dncnn')' for the color image denoising task.

Table 2. Distribution of patchsize in K-QSVD (Kodak24).

Patchsize \ Images, σ		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
$\sigma = 25$	8×8		⋆	⋆							⋆		⋆							⋆			⋆		
	9×9	⋆					⋆	⋆	⋆	⋆		⋆		⋆								⋆			
	10×10				⋆										⋆			⋆	⋆						⋆
	11×11					⋆										⋆	⋆				⋆				
	12×12																							⋆	
$\sigma = 35$	8×8					⋆			⋆				⋆			⋆			⋆			⋆			
	9×9			⋆							⋆						⋆			⋆	⋆		⋆		
	10×10	⋆	⋆		⋆		⋆			⋆				⋆											⋆
	11×11							⋆				⋆			⋆			⋆							
	12×12																							⋆	
$\sigma = 50$	8×8	⋆	⋆		⋆		⋆			⋆	⋆			⋆		⋆				⋆					
	9×9					⋆											⋆				⋆	⋆			⋆
	10×10											⋆						⋆	⋆				⋆		
	11×11							⋆	⋆						⋆										
	12×12			⋆									⋆											⋆	

4.2. Effectiveness of zero constraint and q-TV. The advantages of representing color images with zero constraint can be seen in Fig. 6. We test the images with noise level σ=35 and present the results of several sparse representation-based denoising methods. For the K-SVD method, we test their Matlab code with the default parameters. The K-SVD denoising method [32] (the third column) introduces blurring and color distortion. The K-QSVD$^+$ (the fourth column) represents the best results of different patch sizes and improves the original K-QSVD dictionary learning method. However, there are artifacts shown in some results of the K-QSVD$^+$ method. From the zooming parts, we see clearly that our pQS model is better than those dictionary learning methods, and can eliminate artifacts and avoid color distortion at the same time.

As for the role of the proposed q-TV term, we compare the proposed pQS and pQSTV by the average numerical results on the tested images with different noise levels. As shown in Fig. 2 and Fig. 12, the TV term consistently helps improve the results.

4.3. Experimental results. In this subsection, we give a thorough evaluation by comparing the proposed models to all the competing methods on three benchmark datasets with different noise levels $\sigma = 25, 35, 50$. In Tables 3, 4, and 5, respectively, we list all the PSNR and SSIM values of each image as well as the average results of each method. For a clearer display, we also highlight the best results in bold and underline the second-best ones.

As shown in these tables, the proposed pQSTV and pQS methods get the highest and the second-highest numerical results in most cases. From the average values, one can see that our pQSTV model achieves the best numerical results among all the methods, which proves the superiority of the proposed methods. Among the competing methods, the classical K-SVD method proposed in [32] was first used in gray image processing, and showed good image processing proficiency. From those values in Tables 3, 4, and 5, we know that the K-SVD method is still good at the SSIM values of the denoised images. As an improvement method of dictionary learning, the method PGPD represents color images by vectors and shows better results. Nevertheless, it ignores the inherent relationships among the color channels. Different from these methods, K-QSVD can keep the inner relation of RGB channels by calculating the images in the quaternion domain. This explains why the denoising effects of the K-QSVD and the proposed methods are generally better than other dictionary learning methods.

We further compare the proposed models with other two methods including TV terms, i.e., ℓ_1-ROF [5] and SV-TV [20]. The conventional ℓ_1-ROF model [5] may suffer from the oversmoothness and therefore has better performance in those

TABLE 3. PSNR and SSIM values of different denoising models in noise level σ=25.

images	noisy		ℓ_1-ROF		SV-TV		K-SVD		K-QSVD		K-QSVD$^+$		PGPD		DnCNN		pQS		pQSTV	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Results of Kodak24																				
K01	20.17	0.6294	23.11	0.7062	27.63	0.7955	25.66	0.8217	27.90	0.8653	27.91	0.8829	26.92	0.7456	26.89	0.8570	27.89	0.8722	**28.04**	**0.8908**
K02	20.18	0.8608	28.72	0.9747	29.99	0.7159	30.09	0.9821	31.26	0.9843	31.26	0.9843	31.32	0.7783	30.77	0.9834	31.34	0.9848	**31.68**	**0.9872**
K03	20.17	0.4438	29.89	0.9356	30.32	0.7266	30.78	0.9185	32.46	0.9260	32.46	0.9260	32.53	0.8552	31.45	0.9385	32.44	0.9191	**33.09**	**0.9597**
K04	20.17	0.6235	28.18	0.9469	29.96	0.8242	29.85	0.9505	30.95	0.9575	31.00	0.9648	31.21	0.7975	30.52	0.9586	31.03	0.9555	**31.34**	**0.9675**
K05	20.17	0.6002	23.13	0.7517	27.86	0.8269	24.72	0.8223	27.91	0.8730	28.07	0.8959	27.52	0.8107	27.09	0.8750	28.16	0.8825	**28.25**	**0.9038**
K06	20.17	0.5412	24.62	0.7835	28.41	0.7592	26.57	0.8584	28.89	0.8773	28.89	0.8957	28.33	0.7723	27.78	0.8784	29.12	0.8886	**29.18**	**0.9053**
K07	20.18	0.4983	27.85	0.9104	29.99	0.7860	29.57	0.9269	31.51	0.9360	31.67	0.9516	31.83	0.8948	30.67	0.9372	31.70	0.9356	**32.02**	**0.9589**
K08	20.17	0.6271	21.72	0.7124	27.28	0.8298	23.75	0.8150	27.80	0.8491	27.80	0.8927	27.50	0.8372	26.53	0.8658	27.87	0.8586	**27.89**	**0.8994**
K09	20.17	0.3293	28.13	0.8329	30.16	0.8110	29.54	0.8208	31.72	0.7649	31.88	0.8782	32.02	0.8616	31.24	0.8647	31.95	0.7628	**32.35**	**0.8933**
K10	20.17	0.3401	27.78	0.8321	30.16	0.8249	29.57	0.8432	31.72	0.8131	31.72	0.8131	31.82	0.8421	30.93	0.8705	31.74	0.8169	**32.25**	**0.8981**
K11	20.17	0.4452	26.05	0.8262	29.11	0.7473	27.26	0.8377	29.54	0.7914	29.60	0.8837	29.32	0.7735	28.93	0.8701	29.73	0.7977	**29.88**	**0.8959**
K12	20.17	0.5004	29.64	0.9205	30.25	0.7238	30.32	0.9227	31.84	0.9311	31.84	0.9311	32.05	0.8127	31.21	0.9335	31.78	0.9309	**32.29**	**0.9472**
K13	20.17	0.6270	21.35	0.7041	**26.78**	0.7954	22.49	0.7719	26.12	0.8485	26.13	0.8686	25.17	0.6769	25.20	0.8423	26.18	0.8577	26.21	**0.8745**
K14	20.17	0.5265	25.30	0.7675	28.48	0.7755	26.58	0.8175	28.62	0.7914	28.67	0.8529	28.18	0.7419	28.07	0.8416	28.79	0.8061	**28.83**	**0.8671**
K15	20.18	0.4932	28.69	0.9085	29.96	0.7282	29.93	0.9111	31.02	0.8976	31.08	0.9314	**31.66**	0.8279	30.23	0.9167	31.10	0.8966	31.41	**0.9362**
K16	20.18	0.3672	27.48	0.7492	29.52	0.7316	29.24	0.8134	30.44	0.7655	30.48	0.8416	30.22	0.7791	29.79	0.8282	30.68	0.7864	**31.02**	**0.8666**
K17	20.17	0.3265	27.75	0.8235	30.00	0.8489	28.89	0.8263	31.03	0.7604	31.07	0.8711	30.95	0.8325	30.20	0.8462	31.15	0.7579	**31.35**	**0.8853**
K18	20.17	0.4990	24.59	0.7952	28.45	0.8626	25.77	0.8136	28.32	0.7717	28.37	0.8668	27.77	0.7539	27.37	0.8493	28.51	0.7788	**28.55**	**0.8749**
K19	20.17	0.4680	25.14	0.8537	29.05	0.8353	27.95	0.8689	30.18	0.8358	30.18	0.8358	30.04	0.7950	29.21	0.8970	30.23	0.8391	**30.42**	**0.9203**
K20	20.17	0.4085	28.66	0.9067	30.10	0.7320	29.45	0.9043	31.55	0.7635	31.67	0.9214	31.65	0.8491	28.69	0.9001	31.52	0.7589	**31.84**	**0.9342**
K21	20.17	0.5283	25.42	0.8260	28.95	0.7645	26.47	0.8565	29.42	0.8711	29.51	0.9056	28.83	0.8268	28.63	0.8936	29.53	0.8767	**29.58**	**0.9342**
K22	20.18	0.5200	26.72	0.8673	29.23	0.7457	28.09	0.8759	29.46	0.8593	29.66	0.8593	29.38	0.7504	29.13	0.8980	29.78	0.8590	**29.99**	**0.9127**
K23	20.17	0.5729	29.89	0.9451	30.70	0.7424	30.66	0.9412	33.04	0.9431	33.12	0.9684	33.38	0.8848	32.15	0.9546	32.85	0.9379	**33.66**	**0.9697**
K24	20.17	0.4481	23.79	0.7596	28.30	0.7836	25.07	0.8011	28.43	0.7863	28.51	0.8612	28.04	0.7918	27.42	0.8451	28.64	0.8000	**28.68**	**0.8728**
Aver.	20.17	0.5094	26.40	0.8350	29.19	0.7799	27.84	0.8634	30.06	0.8526	30.11	0.8952	29.90	0.8038	29.17	0.8529	30.15	0.8567	**30.41**	**0.9148**
Results of Set5																				
S01	20.18	0.6059	29.13	**0.9057**	30.20	0.8075	29.89	0.7886	30.56	0.8281	30.58	0.8279	30.76	0.8324	30.16	0.8129	30.80	0.8398	**31.03**	0.8412
S02	20.47	0.8283	23.14	0.9211	28.16	0.8479	25.33	0.9467	28.64	0.9708	28.73	0.9043	28.28	0.9072	28.33	0.9689	28.85	0.9037	**28.86**	**0.9778**
S03	20.18	0.6832	28.13	**0.8042**	28.38	0.6778	26.51	0.5739	28.41	0.6499	28.43	0.6505	28.69	0.6384	28.60	0.6628	28.56	0.6637	**28.80**	0.6619
S04	20.18	0.4881	27.45	0.9116	28.92	0.7766	26.76	0.7057	29.06	0.7858	29.08	0.7866	**30.39**	0.8281	29.45	0.7971	29.58	0.8019	30.12	**0.9122**
S05	20.18	0.6244	26.28	0.9042	29.39	0.8127	26.45	0.7570	30.09	0.8676	30.03	0.8680	30.21	0.8817	29.49	0.8522	30.31	0.8730	**30.57**	**0.9094**
Aver.	20.33	0.7140	27.75	**0.8814**	29.22	0.7687	26.99	0.7544	29.35	0.8204	29.37	0.8075	29.67	0.8176	29.43	0.7813	29.62	0.8164	**29.88**	0.8605
Results of CSet8																				
C01	20.38	0.4340	24.50	0.7613	27.77	0.7610	25.52	0.8077	29.09	0.8829	29.13	0.8599	28.91	0.8575	28.61	0.8702	29.26	0.8639	**29.28**	**0.8715**
C02	20.31	0.7689	21.68	0.7361	24.79	0.7791	22.13	0.7614	25.27	0.8879	25.14	0.7842	24.66	0.7406	24.29	0.8518	**25.39**	0.8052	25.29	**0.8988**
C03	20.33	0.6513	24.31	0.8340	27.41	0.7958	26.33	0.8793	28.72	0.9153	28.76	0.8465	28.43	0.8345	27.62	0.8991	28.89	0.8539	**28.97**	**0.9244**
C04	20.44	0.5536	23.65	0.7922	27.69	0.7676	25.12	0.8358	28.34	0.8952	28.23	0.8232	27.88	0.8177	27.64	0.8850	28.42	0.8287	**28.66**	**0.9255**
C05	20.47	0.8283	23.14	0.9211	28.16	0.8479	25.33	0.9467	28.64	0.9708	28.73	0.9043	28.28	0.9072	28.33	0.9689	**28.85**	0.9037	28.84	**0.9778**
C06	20.30	0.6789	27.80	0.9272	28.53	0.7039	30.30	0.9534	31.55	0.9638	31.65	0.8206	**32.34**	0.8251	31.14	0.9601	31.59	0.8211	32.16	**0.9783**
C07	20.40	0.8271	25.80	0.9480	28.24	0.7665	27.06	0.9589	29.88	0.9782	29.86	0.8520	29.68	0.8519	29.13	0.9738	30.04	0.8573	**30.18**	**0.9826**
C08	20.49	0.8453	25.79	0.9515	27.26	0.7470	26.93	0.9611	28.92	0.9739	29.07	0.8297	29.51	0.8358	29.13	**0.9743**	29.33	0.8322	**29.66**	0.8349
Aver.	20.39	0.6984	24.58	0.8589	27.48	0.7711	26.09	0.8880	28.80	0.9335	28.82	0.8401	28.71	0.8337	28.24	0.9229	28.97	0.8458	**29.13**	**0.9242**
Average results of all three databases																				
Aver.	20.30	0.6202	26.24	0.8584	28.63	0.7732	26.97	0.8353	29.40	0.8688	29.43	0.8476	29.43	0.8184	28.95	0.8524	29.58	0.8396	**29.80**	**0.8998**

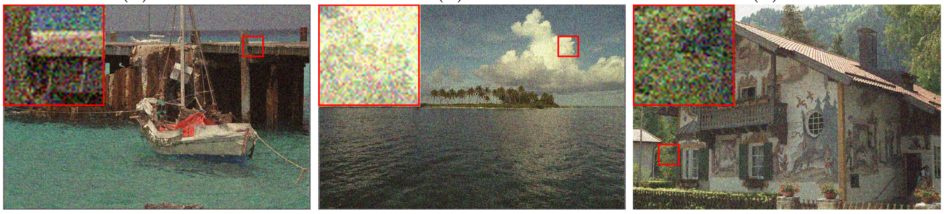
TABLE 4. PSNR and SSIM values of different denoising models in noise level σ=35.

images	noisy		ℓ_1-ROF		SV-TV		K-SVD		K-QSVD		K-QSVD$^+$		PGPD		DnCNN		pQS		pQSTV	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Results of Kodak24																				
K01	17.25	0.4860	22.55	0.6752	25.68	0.7117	25.01	0.7949	26.24	0.8086	26.25	0.8311	25.51	0.6719	25.43	0.8068	26.48	0.8252	**26.55**	**0.8481**
K02	17.25	0.7668	28.09	0.9706	27.77	0.6029	28.69	0.9756	29.68	0.9780	29.76	0.9799	30.20	0.7435	29.09	0.9756	29.73	0.9785	**30.42**	**0.9825**
K03	17.26	0.3405	29.12	0.9221	28.00	0.6084	29.33	0.8778	30.75	0.8969	30.77	0.9291	31.28	0.8288	29.79	0.9117	30.79	0.8871	**31.46**	**0.9452**
K04	17.26	0.4871	27.49	0.9351	27.74	0.7294	28.54	0.9281	29.38	0.9437	29.43	0.9505	29.96	0.7604	29.00	0.9429	29.52	0.9410	**29.97**	**0.9577**
K05	17.25	0.4656	22.44	0.7147	25.78	0.7486	24.18	0.7906	26.24	0.8190	26.24	0.8190	25.78	0.7450	25.25	0.8206	26.43	0.8334	**26.51**	**0.8625**
K06	17.25	0.3995	24.12	0.7624	26.35	0.6599	25.81	0.8249	27.11	0.8253	27.12	0.8508	26.90	0.7117	26.18	0.8360	**27.48**	0.8425	**27.59**	**0.8702**
K07	17.25	0.3672	26.66	0.8828	27.65	0.6850	28.23	0.8881	29.73	0.9092	29.75	0.9283	30.08	0.8646	28.73	0.9060	30.05	0.9080	**30.21**	**0.9421**
K08	17.25	0.5020	21.10	0.6763	**26.75**	0.8136	23.34	0.7840	26.01	0.7923	26.01	0.7923	25.74	0.7837	24.78	0.8168	26.18	0.8101	26.19	**0.8602**
K09	17.25	0.2317	27.35	0.8050	27.81	0.7098	28.21	0.7431	29.90	0.6925	30.18	0.8343	30.58	0.8351	29.59	0.8134	30.18	0.6888	**30.80**	**0.8610**
K10	17.24	0.2266	27.08	0.8040	27.82	0.7274	28.24	0.7778	29.74	0.7492	29.92	0.8394	30.33	0.8097	29.27	0.8258	30.14	0.7575	**30.59**	**0.8663**
K11	17.25	0.3204	25.45	0.8027	26.98	0.6455	26.42	0.7866	28.00	0.7433	28.01	0.8440	27.93	0.7253	27.31	0.8241	28.18	0.7498	**28.35**	**0.8639**
K12	17.26	0.3528	28.80	0.9098	27.95	0.6125	28.90	0.8908	30.43	0.9115	30.43	0.9115	30.91	0.7842	29.69	0.9120	30.32	0.9090	**30.96**	**0.9318**
K13	17.25	0.4895	20.96	0.6789	**24.90**	0.7173	22.15	0.7466	24.52	0.7896	24.55	0.8159	23.69	0.5980	23.68	0.7905	24.70	0.8067	**24.90**	**0.8286**
K14	17.25	0.4059	24.58	0.7383	26.53	0.6871	25.80	0.7776	26.94	0.7280	27.00	0.8019	26.81	0.6842	26.53	0.7941	27.20	0.7484	**27.29**	**0.8211**
K15	17.26	0.3689	27.92	0.8934	27.71	0.6149	28.77	0.8787	29.53	0.8680	29.53	0.8680	**30.46**	0.8013	28.28	0.8839	29.52	0.8635	29.80	**0.9182**
K16	17.25	0.2446	27.00	0.7273	27.36	0.6210	28.04	0.7517	28.78	0.6903	28.83	0.7861	28.96	0.7294	28.35	0.7747	29.06	0.7165	**29.40**	**0.8168**
K17	17.25	0.2226	26.98	0.7895	27.71	0.7629	27.86	0.8165	29.38	0.6974	29.44	0.8276	29.50	0.7971	28.26	0.7866	29.43	0.6903	**29.83**	**0.8449**
K18	17.25	0.3823	24.05	0.7697	26.44	0.7867	25.14	0.7680	26.65	0.7115	26.65	0.7115	26.27	0.6882	25.87	0.8007	**26.93**	0.7224	26.89	**0.8288**
K19	17.25	0.3522	24.51	0.8335	26.87	0.7447	27.03	0.7948	28.66	0.7977	28.78	0.8851	28.87	0.7629	27.74	0.8604	28.76	0.7913	**29.05**	**0.8937**
K20	17.25	0.2922	27.77	0.8884	27.78	0.6171	28.40	0.8630	29.98	0.7312	30.02	0.8941	30.41	0.8274	26.39	0.8626	29.93	0.7258	**30.46**	**0.9090**
K21	17.25	0.3901	24.78	0.8022	26.78	0.6633	25.68	0.8159	27.80	0.8279	27.80	0.8279	27.34	0.7835	27.02	0.8539	27.90	0.8347	**28.12**	**0.8797**
K22	17.25	0.3873	26.08	0.8492	27.16	0.6449	27.09	0.8346	28.17	0.8178	28.20	0.8775	28.17	0.7002	27.78	0.8654	28.26	0.8166	**28.66**	**0.8921**
K23	17.24	0.4529	28.89	0.9322	28.25	0.6253	29.27	0.9130	31.57	0.9262	31.58	0.9580	31.94	0.8650	30.30	0.9363	31.32	0.9173	**31.97**	**0.9596**
K24	17.25	0.3281	23.31	0.7278	26.24	0.6909	24.48	0.7492	26.67	0.7192	26.71	0.8070	26.42	0.7325	25.71	0.7920	26.85	0.7328	**26.96**	**0.8252**
Aver.	17.25	0.3860	25.42	0.8121	27.08	0.6846	26.86	0.8238	28.41	0.8073	28.46	0.8571	28.50	0.7597	27.50	0.8497	28.56	0.8124	**28.87**	**0.8837**
Results of Set5																				
S01	17.24	0.4881	27.35	0.8738	27.88	0.7363	27.79	0.7101	28.20	0.7766	28.21	0.7763	28.85	0.7955	28.09	0.7601	28.56	0.7958	**28.98**	**0.8928**
S02	17.82	0.7415	22.56	0.9073	26.00	0.7840	24.56	0.9359	26.41	0.9537	26.40	0.8649	26.43	0.8753	26.53	0.9545	27.05	0.8752	**27.18**	**0.9632**
S03	17.26	0.5765	26.55	**0.7482**	26.35	0.5879	26.51	0.5739	26.62	0.5785	26.62	0.5788	**27.27**	0.5806	26.99	0.6007	26.77	0.5904	27.11	0.6323
S04	17.22	0.3580	25.99	0.8749	26.63	0.6922	26.76	0.7057	26.49	0.7079	26.46	0.7076	**28.02**	0.7658	27.13	0.7244	26.95	0.7256	27.38	**0.8934**
S05	17.24	0.4974	25.21	0.8684	26.86	0.7266	26.45	0.7570	27.62	0.8158	27.55	0.8151	28.11	0.8415	27.38	0.7953	28.02	0.8276	**28.47**	**0.9199**
Aver.	17.53	0.6110	26.28	0.8413	26.93	0.6858	26.41	0.7365	27.07	0.7665	27.05	0.7485	27.74	0.7717	27.40	0.7201	27.47	0.7629	**27.82**	**0.8603**
Results of CSet8																				
C01	17.77	0.3464	23.98	0.7032	25.95	0.6819	24.78	0.7435	26.80	0.8345	26.88	0.8101	27.08	0.8178	26.86	0.8271	27.44	0.8278	**27.54**	**0.8303**
C02	17.52	0.6530	21.39	0.7220	23.28	0.7075	21.69	0.7400	23.33	0.8195	23.29	0.6707	22.87	0.6265	22.72	0.7897	23.68	0.7140	**23.85**	**0.7970**
C03	17.55	0.5264	23.80	0.8082	25.54	0.7168	25.40	0.8513	26.64	0.8777	26.72	0.7844	26.74	0.7859	26.06	0.8670	27.13	0.8020	**27.14**	**0.8812**
C04	17.62	0.4403	23.11	0.7459	25.51	0.6760	24.20	0.7837	25.88	0.8438	25.90	0.7470	25.95	0.7577	25.86	0.8431	26.38	0.7680	**26.53**	**0.8465**
C05	17.82	0.7415	22.56	0.9073	26.00	0.7840	24.56	0.9359	26.41	0.9537	26.40	0.8649	26.43	0.8753	26.53	**0.9545**	27.05	0.8752	**27.13**	0.9122
C06	17.49	0.5366	26.73	0.9067	26.58	0.6108	28.74	0.9355	29.63	**0.9489**	29.83	0.7930	30.95	0.8053	29.54	0.9457	30.13	0.7959	**30.98**	0.9442
C07	17.61	0.7189	25.06	0.9480	26.21	0.6751	25.93	0.9463	27.54	**0.9640**	27.57	0.7959	27.70	0.8046	27.37	0.9611	28.08	0.8102	**28.22**	0.9626
C08	17.75	0.7499	24.93	0.9398	25.55	0.6632	25.80	0.9488	26.79	0.9582	26.63	0.7762	27.61	0.7964	27.29	0.9615	27.14	0.7851	**27.71**	**0.9758**
Aver.	17.64	0.5891	23.94	0.8351	25.58	0.6894	25.14	0.8606	26.63	0.9000	26.65	0.7803	26.92	0.7837	26.53	0.8937	27.13	0.7973	**27.39**	**0.9050**
Average results of all three databases																				
Aver.	17.47	0.5287	25.21	0.8295	26.53	0.7002	26.14	0.8070	27.37	0.8246	27.39	0.7953	27.72	0.7717	27.14	0.8212	27.72	0.7908	**28.03**	**0.8830**

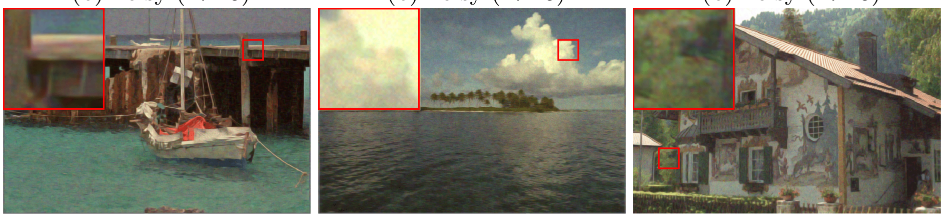
TABLE 5. PSNR and SSIM values of different denoising models in noise level σ=50.

images	noisy		ℓ_1-ROF		SV-TV		K-SVD		K-QSVD		K-QSVD$^+$		PGPD		DnCNN		pQS		pQSTV	
Results of Kodak24																				
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
K01	14.16	0.3364	21.94	0.6416	23.54	0.6034	23.93	0.7448	24.39	0.7560	24.39	0.7560	24.89	0.7867	23.86	0.7404	24.08	0.5828	**25.13**	**0.7958**
K02	14.15	0.6276	27.39	0.9653	25.24	0.4645	26.91	0.9641	28.13	0.9703	28.13	0.9703	**29.05**	0.7099	26.85	0.9607	28.24	0.9718	28.93	**0.9762**
K03	14.15	0.2595	28.14	0.9034	25.37	0.4648	27.44	0.8128	28.82	0.8979	28.97	0.9184	**29.95**	0.7993	27.70	0.8652	28.89	0.8862	29.81	**0.9310**
K04	14.16	0.3464	26.70	0.9205	25.22	0.6014	26.81	0.8889	27.91	0.9329	27.91	0.9329	**28.73**	0.7232	27.13	0.9146	27.98	0.9280	28.51	**0.9406**
K05	14.15	0.3283	21.69	0.6723	23.57	0.6427	23.25	0.7366	24.21	0.7722	24.22	0.7723	24.07	0.662	23.31	0.7449	24.64	0.7968	**24.77**	**0.8014**
K06	14.15	0.2792	23.56	0.7389	24.11	0.5378	24.67	0.7729	25.36	0.7962	25.36	0.7962	25.47	0.6399	24.48	0.7808	25.75	0.8138	**25.92**	**0.8232**
K07	14.15	0.2463	25.47	0.8467	25.05	0.5553	26.39	0.8187	27.39	0.8837	27.46	0.8841	28.24	0.8219	26.55	0.8510	27.72	0.8833	**28.36**	**0.9129**
K08	14.16	0.3679	20.36	0.6327	23.02	0.6662	22.54	0.7320	24.03	0.7771	24.05	0.7760	24.03	0.7200	22.84	0.7484	24.40	0.7987	**24.51**	**0.8083**
K09	14.16	0.1543	26.44	0.7692	25.20	0.5773	26.50	0.6362	28.32	0.7759	28.32	0.7759	**29.04**	0.7973	27.67	0.7350	28.49	0.7697	29.02	**0.8150**
K10	14.15	0.1397	26.27	0.7688	25.21	0.5921	26.47	0.6795	28.16	0.7884	28.16	0.7884	28.70	0.7658	27.32	0.7566	28.34	0.7847	**28.79**	**0.8186**
K11	14.15	0.2090	24.73	0.7700	24.58	0.5192	25.12	0.7107	26.41	0.6891	26.41	0.7951	26.49	0.6703	25.40	0.7534	26.65	0.6928	**26.83**	**0.8191**
K12	14.16	0.2217	27.80	0.8941	25.34	0.4751	27.10	0.8366	28.69	0.8812	28.73	0.8968	**29.72**	0.7529	27.74	0.8766	28.69	0.8760	29.36	**0.9100**
K13	14.15	0.3454	20.49	0.6501	22.83	0.6147	21.57	0.7051	22.90	0.7201	22.90	0.7201	22.30	0.5106	22.10	0.7213	23.11	0.7400	**23.15**	**0.7718**
K14	14.15	0.2923	23.79	0.7036	24.30	0.5716	24.67	0.7210	25.30	0.6608	25.45	0.7466	25.44	0.6196	24.78	0.7328	25.63	0.6809	**25.86**	**0.7680**
K15	14.15	0.2552	26.98	0.8714	25.15	0.4766	27.17	0.8251	27.97	0.8295	27.97	0.8295	**29.17**	0.7699	25.78	0.8332	27.92	0.8208	28.42	**0.8847**
K16	14.15	0.1481	26.40	0.6996	24.94	0.4871	26.43	0.6595	27.25	0.6072	27.27	0.7218	27.75	0.6748	26.80	0.7063	27.48	0.6268	**27.85**	**0.7551**
K17	14.15	0.1390	26.04	0.7422	25.13	0.6410	26.34	0.6768	27.50	0.6050	27.64	0.7685	27.87	0.7509	25.87	0.7009	27.78	0.6020	**28.06**	**0.7900**
K18	14.14	0.2656	23.43	0.7376	25.15	0.6750	24.11	0.7062	25.01	0.6444	25.01	0.7677	24.84	0.6118	24.15	0.7338	25.32	0.6544	**25.36**	**0.7820**
K19	14.14	0.2422	23.83	0.8020	24.43	0.6208	25.64	0.7419	27.25	0.6072	27.25	0.6072	**27.57**	0.7260	25.92	0.7997	27.48	0.6268	27.59	**0.8623**
K20	14.16	0.1919	26.76	0.8580	25.16	0.4794	27.01	0.7987	28.25	0.6942	28.37	0.8547	**29.02**	0.8017	23.75	0.8071	28.23	0.6840	28.88	**0.8800**
K21	14.16	0.2603	24.07	0.7747	24.41	0.5380	24.54	0.7530	25.89	0.7680	25.98	0.8189	25.86	0.7281	25.27	0.7970	26.19	0.7793	**26.36**	**0.8387**
K22	14.15	0.2593	25.38	0.8249	24.79	0.5189	25.70	0.7746	26.71	0.7708	26.73	0.8410	26.95	0.6489	26.20	0.8181	26.81	0.7682	**27.12**	**0.8596**
K23	14.16	0.3323	27.79	0.9130	25.51	0.4810	27.46	0.8655	29.64	0.8979	29.73	**0.9420**	**30.40**	0.8397	27.92	0.9032	29.54	0.8882	30.17	0.9376
K24	14.16	0.2203	22.76	0.6885	23.96	0.5689	23.59	0.6706	24.96	0.6391	24.98	0.7375	24.89	0.6606	23.96	0.7175	25.19	0.6552	**25.31**	**0.7557**
Aver.	14.15	0.2695	24.93	0.7829	24.63	0.5572	25.47	0.7597	26.69	0.7652	26.72	0.8091	27.10	0.7164	25.56	0.7916	26.86	0.7630	**27.25**	**0.8427**
Results of Set5																				
S01	14.16	0.3574	24.89	0.8192	25.17	0.6386	25.00	0.5964	25.38	0.7134	25.40	0.7136	**26.52**	0.7479	25.66	0.6919	25.75	0.7305	26.32	**0.8211**
S02	15.14	0.6241	21.50	0.8794	23.52	0.7047	23.08	0.9103	23.31	0.9162	23.08	0.7877	23.99	0.8258	24.24	**0.9264**	24.07	0.8166	**24.39**	0.8983
S03	14.15	0.4444	24.25	0.6743	23.90	0.4839	23.99	0.4731	24.54	0.5172	24.54	0.5171	25.33	0.5204	24.75	0.5250	24.63	0.5225	**25.34**	**0.6817**
S04	14.16	0.2405	23.78	0.8176	23.92	0.5824	23.95	0.5948	23.48	0.6096	23.45	0.6088	**25.19**	0.6802	24.47	0.6297	23.95	0.6298	24.58	**0.8329**
S05	14.15	0.3585	23.52	0.8091	24.21	0.6275	24.22	0.6535	24.48	0.7341	24.49	0.7353	25.31	0.7846	24.90	0.7176	25.05	0.7569	**25.77**	**0.8099**
Aver.	14.64	0.4870	24.11	0.7801	24.30	0.5831	24.05	0.6456	24.24	0.6981	24.19	0.6725	25.27	0.7118	24.95	0.6411	24.69	0.6913	**25.28**	**0.8088**
Results of CSet8																				
C01	15.14	0.2573	22.95	0.6224	23.82	0.5907	23.33	0.6331	23.96	0.7545	23.97	0.7240	24.68	0.7552	**24.79**	0.7612	24.63	0.7493	24.77	**0.7616**
C02	14.72	0.5156	20.81	0.6944	21.63	0.6153	20.92	0.7000	21.06	0.6972	21.06	0.4850	21.08	0.4783	21.36	0.7212	21.70	0.5614	**21.78**	**0.7606**
C03	14.77	0.3935	22.83	0.7588	23.41	0.6202	23.80	0.7947	23.85	0.8073	23.83	0.6715	24.69	0.7189	24.22	**0.8141**	24.65	0.7114	**24.79**	0.7773
C04	14.76	0.3250	22.13	0.6740	23.18	0.5743	22.83	0.7010	23.11	0.7652	23.08	0.6424	23.81	0.6810	23.89	0.7742	23.74	0.6775	**23.90**	**0.7279**
C05	15.14	0.6241	21.50	0.8794	23.52	0.7047	23.08	0.9103	23.31	0.9162	23.08	0.7877	23.99	0.8258	24.24	**0.9264**	24.07	0.8166	**24.39**	0.8983
C06	14.72	0.3840	25.12	0.8676	24.41	0.5104	26.51	0.8983	26.74	0.9140	26.67	0.7397	28.05	0.7797	27.42	0.9189	27.62	0.7561	**28.24**	**0.9485**
C07	14.83	0.5765	23.79	0.9159	23.98	0.5730	24.26	0.9221	24.61	0.9335	24.67	0.7102	25.51	0.7430	25.33	0.9397	25.48	0.7417	**25.52**	**0.9563**
C08	15.05	0.6227	23.49	0.9177	23.52	0.5703	23.98	0.9254	23.70	0.9234	23.82	0.6990	25.07	0.7391	25.04	0.9201	24.40	0.7187	**25.11**	**0.9285**
Aver.	14.89	0.4623	22.83	0.7913	23.43	0.5949	23.59	0.8106	23.79	0.8389	23.77	0.6824	24.74	0.7151	24.54	0.8495	24.54	0.7166	**24.81**	**0.8561**
Average results of all three databases																				
Aver.	17.47	0.5287	23.96	0.7848	24.12	0.5784	24.37	0.7386	24.91	0.7674	24.89	0.7213	25.70	0.7144	25.02	0.7607	25.36	0.7236	**25.78**	**0.8359**

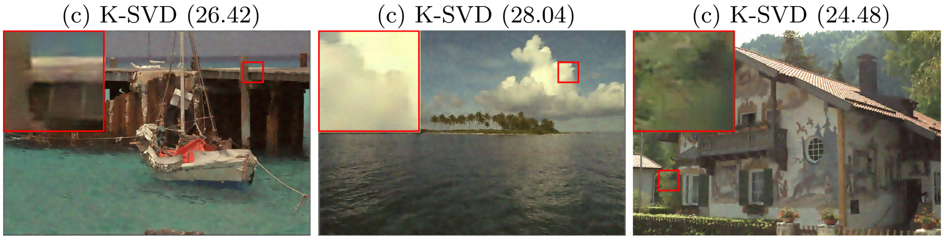
(a) K11 (a) K16 (a) K24



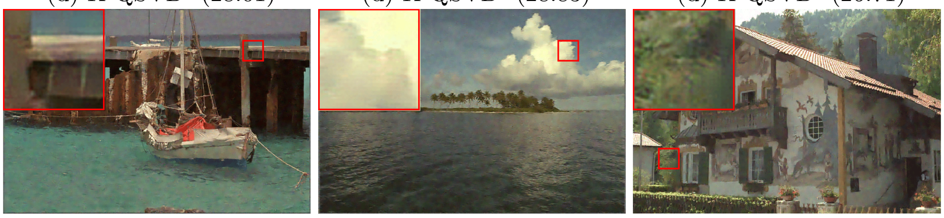
(b) noisy (17.25) (b) noisy (17.25) (b) noisy (17.25)



(c) K-SVD (26.42) (c) K-SVD (28.04) (c) K-SVD (24.48)



(d) K-QSVD$^+$(28.01) (d) K-QSVD$^+$(28.83) (d) K-QSVD$^+$(26.71)

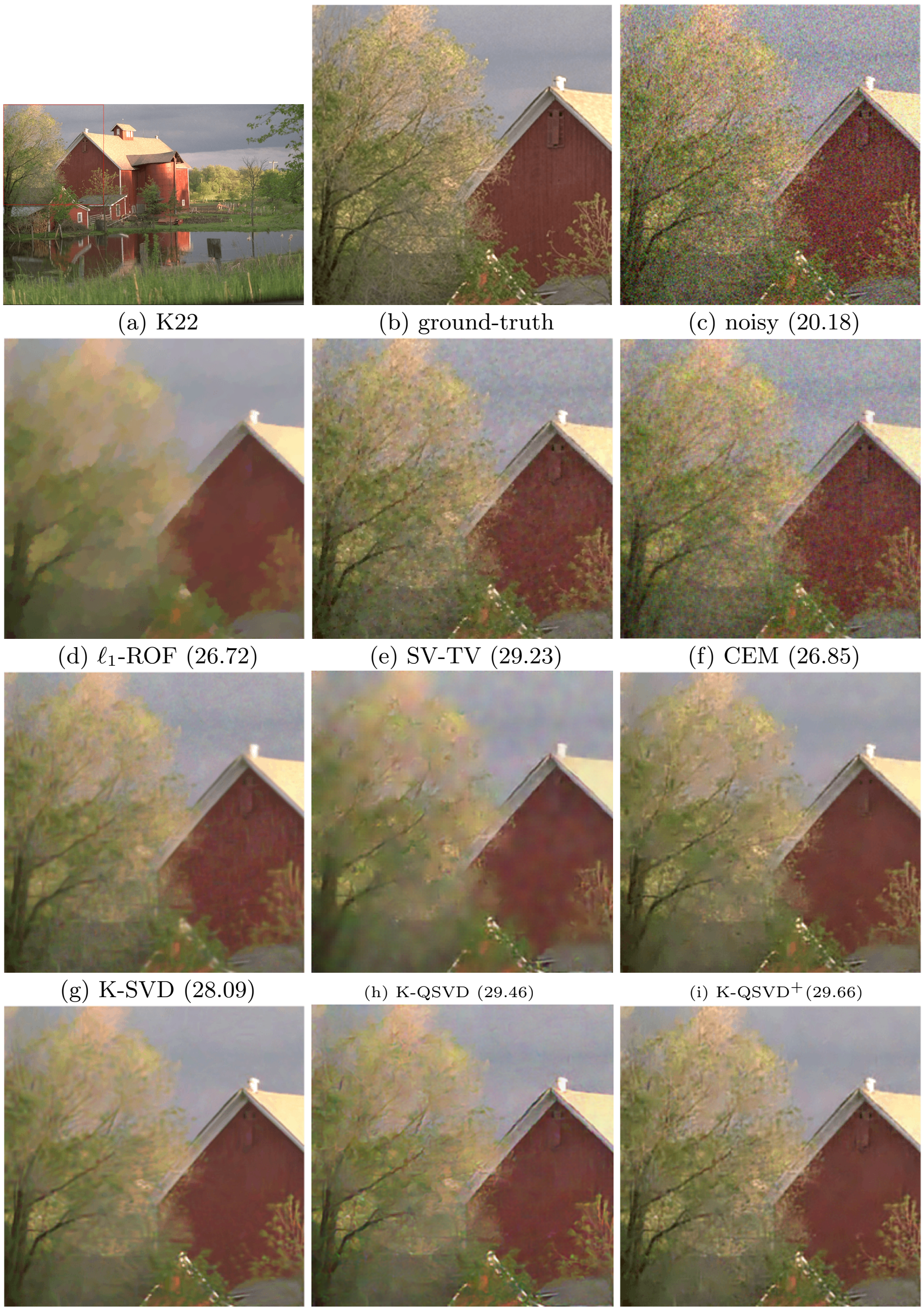


(e) pQS (28.18) (e) pQS (29.06) (e) pQS (26.85)

FIGURE 6. Color image denoising results on K11, K16, and K24. (a) Original image; (b) Noisy image corrupted by Gaussian noise with variance $\sigma = 35$; The denoised image reconstructed by: (c) K-SVD [32], (d) K-QSVD$^+$ [50], (e) the proposed pQS method.

TABLE 6. Average runtime (in seconds without training) for color image denoising.

datasets\methods	ℓ_1-ROF	SV-TV	K-SVD	K-QSVD	K-QSVD$^+$	PGPD	DnCNN	pQS	pQSTV
Kodak24	32.61	14.17	75.92	72.31	89.54	140.05	13.67	101.89	96.23
Set5	8.07	8.64	42.49	20.34	25.41	35.61	5.19	25.55	15.59
CSet8	3.08	6.77	19.00	15.63	21.10	19.03	2.51	21.22	13.44



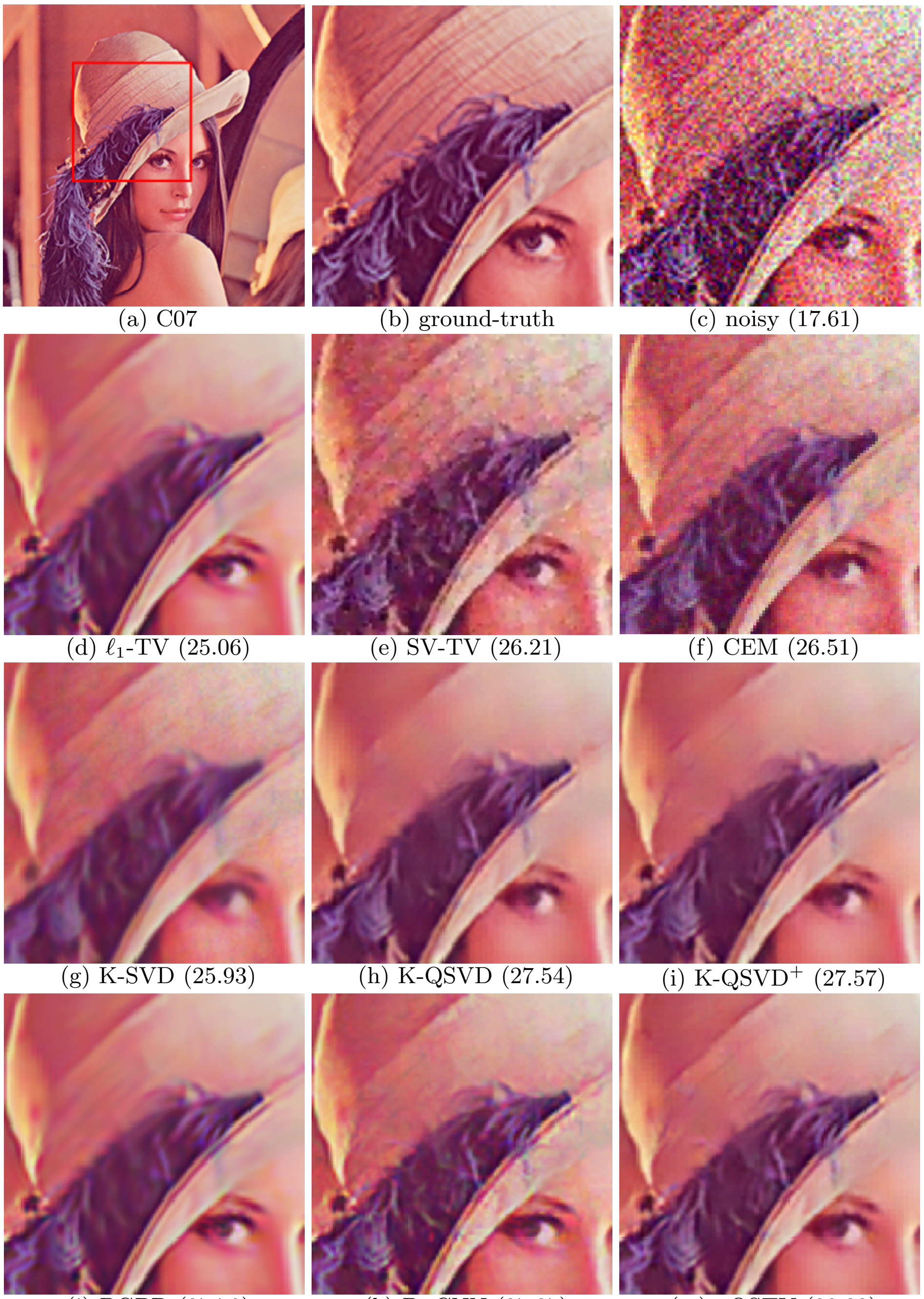
(a) K22 (b) ground-truth (c) noisy (20.18)

(d) ℓ_1-ROF (26.72) (e) SV-TV (29.23) (f) CEM (26.85)

(g) K-SVD (28.09) (h) K-QSVD (29.46) (i) K-QSVD$^+$(29.66)

(j) PGPD (29.38) (k) DnCNN (29.13) (m) pQSTV (29.99)

FIGURE 7. Color image denoising results on K22. (a) Original image; (b) Zooming part of the original image; (c) Noisy image corrupted by Gaussian noise with variance σ= 25; The denoised image reconstructed by: (d) ℓ_1-ROF [5], (e) SV-TV [20], (f) CEM [27], (g) K-SVD [32], (h) K-QSVD[50], (i) K-QSVD$^+$[50], (j) PG-PD [48], (k) DnCNN [56], (m) our pQSTV.



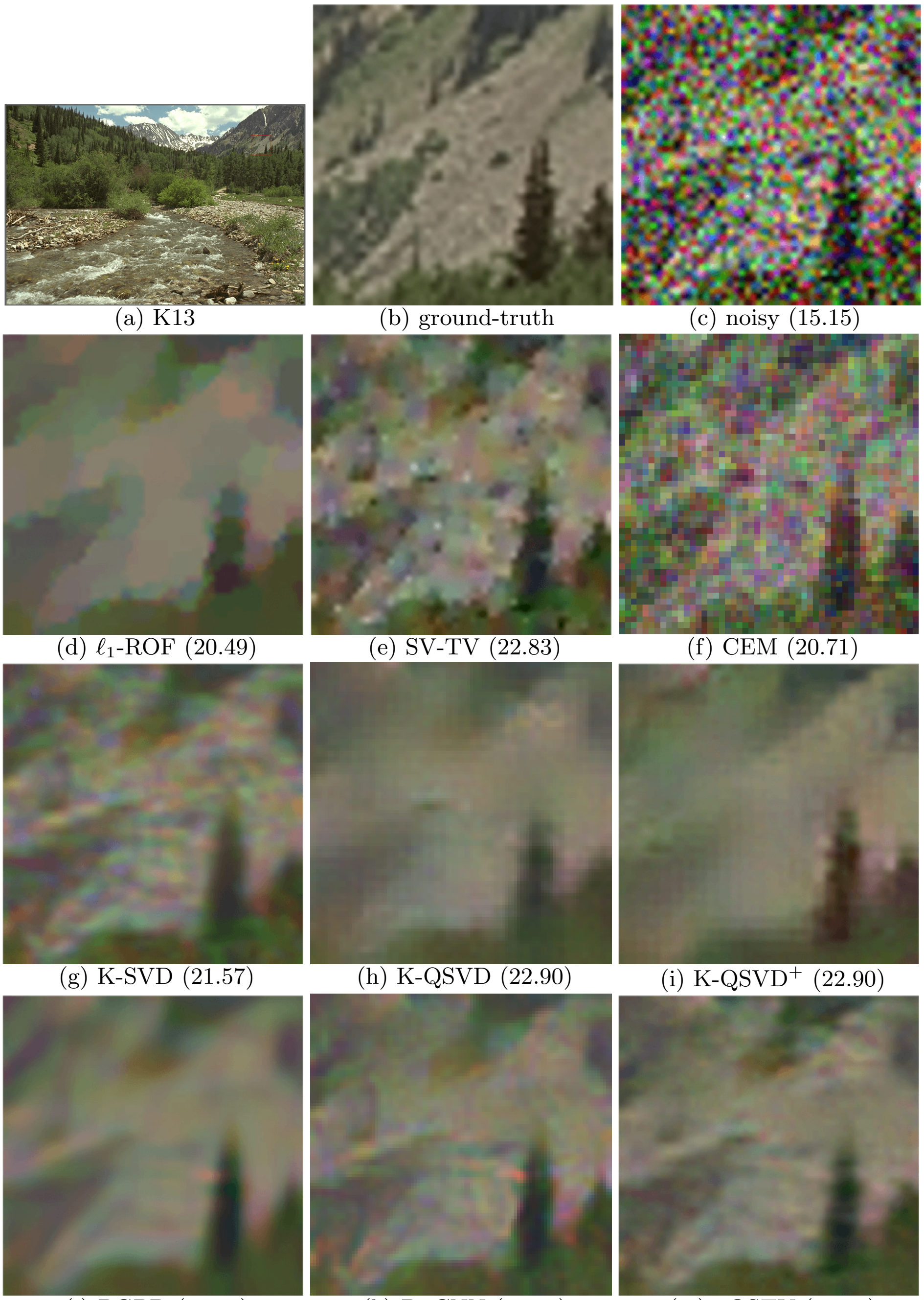
(a) C07 (b) ground-truth (c) noisy (17.61)

(d) ℓ_1-TV (25.06) (e) SV-TV (26.21) (f) CEM (26.51)

(g) K-SVD (25.93) (h) K-QSVD (27.54) (i) K-QSVD$^+$ (27.57)

(j) PGPD (27.70) (k) DnCNN (27.37) (m) pQSTV (28.22)

FIGURE 8. Color image denoising results on C07. (a) Original image; (b) Zooming part of the original image; (c) Noisy image corrupted by Gaussian noise with variance σ= 35; The denoised image reconstructed by: (d) ℓ_1-ROF [5], (e) SV-TV [20], (f) CEM [27], (g) K-SVD [32], (h) K-QSVD[50], (i) K-QSVD$^+$[50], (j) PG-PD [48], (k) DnCNN [56], (m) our pQSTV.



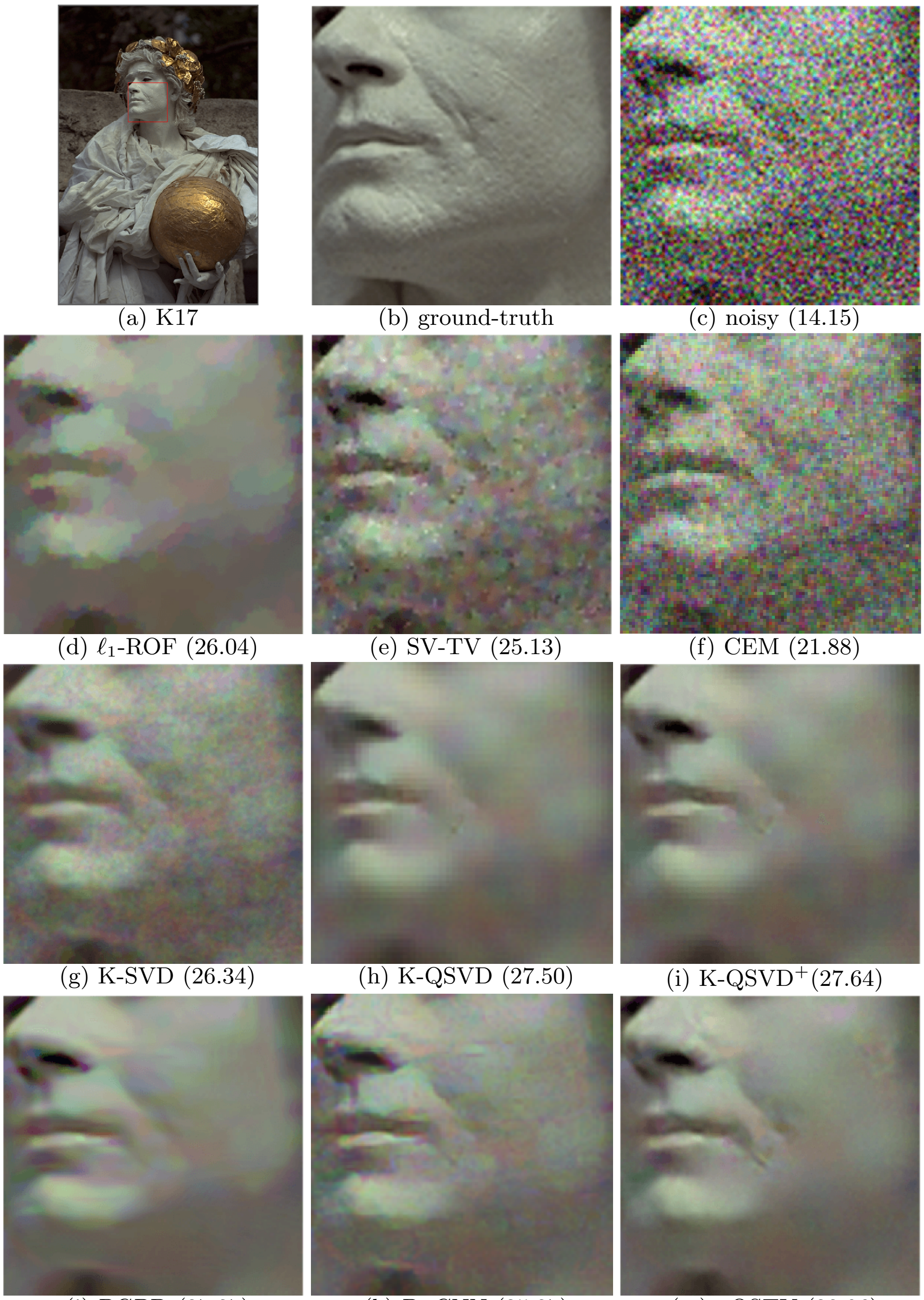
(a) K13 (b) ground-truth (c) noisy (15.15)

(d) ℓ_1-ROF (20.49) (e) SV-TV (22.83) (f) CEM (20.71)

(g) K-SVD (21.57) (h) K-QSVD (22.90) (i) K-QSVD$^+$ (22.90)

(j) PGPD (22.30) (k) DnCNN (22.10) (m) pQSTV (23.15)

FIGURE 9. Color image denoising results on K13. (a) Original image; (b) Zooming part of the original image; (c) Noisy image corrupted by Gaussian noise with variance σ= 50; The denoised image reconstructed by: (d) ℓ_1-ROF [5], (e) SV-TV [20], (f) CEM [27], (g) K-SVD [32], (h) K-QSVD[50], (i) K-QSVD$^+$[50], (j) PG-PD [48], (k) DnCNN [56], (m) our pQSTV.



(a) K17 (b) ground-truth (c) noisy (14.15)

(d) ℓ_1-ROF (26.04) (e) SV-TV (25.13) (f) CEM (21.88)

(g) K-SVD (26.34) (h) K-QSVD (27.50) (i) K-QSVD$^+$(27.64)

(j) PGPD (27.87) (k) DnCNN (25.87) (m) pQSTV (28.06)

FIGURE 10. Color image denoising results on K17. (a) Original image; (b) Zooming part of the original image; (c) Noisy image corrupted by Gaussian noise with variance σ= 50; The denoised image reconstructed by: (d) ℓ_1-ROF [5], (e) SV-TV [20], (f) CEM [27], (g) K-SVD [32], (h) K-QSVD[50], (i) K-QSVD$^+$[50], (j) PG-PD [48], (k) DnCNN [56], (m) our pQSTV.

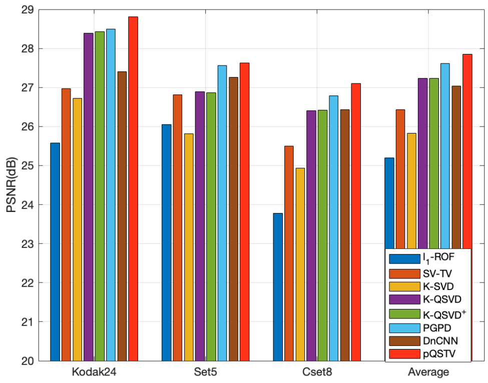


FIGURE 11. Quantitative evaluations on datasets Kodak24, Set5, and CSet8. Our method performs competitively against the state-of-the-art methods.

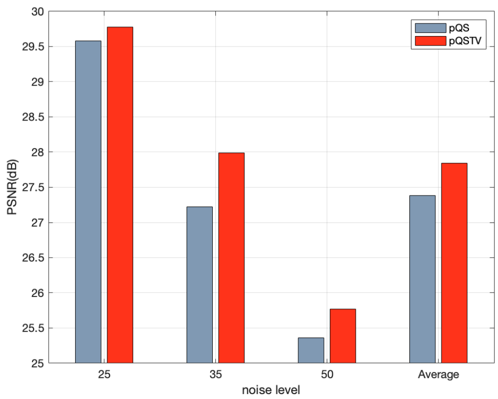


FIGURE 12. Quantitative results of our method pQS and pQSTV with different noise levels. The total variation prior consistently helps improve the results.

images with less texture. As a result of independent process to each channel of RGB, ℓ_1-ROF model also brings about the color distortion to the restored color images. Contrastly, the SV-TV method handles images in HSV (Hue, Saturation, Value) space instead of traditional RGB space. As shown in Fig. 7(e), Fig. 8(e), and Fig. 9(e) the SV-TV model eliminates color distortion in image denoising to some extent when noise level σ is less than or equal to 25. However, when noise levels go to 35 and 50, the results of the SV-TV method become unsatisfactory. Compared to these methods, the proposed methods have a stable and pleasing performance in different noise levels in terms of both noise removal and texture reservation abilities. For the denoising results of CEM [27], we can see that CEM has a competitive denoising performance for images with a low noise level, but for images with a high noise level, there still has room for improvement.

We also compare the proposed models with a learning-based denoising method DnCNN [56], which trains a deep neural network by a huge number of data beforehand. Compared to this data-driven network, our model computes a specific dictionary for any given noisy image and therefore produces more reliable results. The numerical results in the table also show that our methods are better than D-nCNN. For all the methods compared, we list the average running time in Table 6. Here, the training time of the network and the dictionary are not considered.

Lastly, we exhibit some examples with different noise levels in Figs. 7, 8, 9, and 10 for visual comparison. To get a better observation, we reveal the zooming parts of the denoised images in the figures. As shown in the zooming parts, some noise spots still remain in the images denoised by K-SVD and SV-TV methods. Tuning to the best overlapping patch size for K-QSVD, the K-QSVD$^+$ method removes Gaussian noise completely, however it introduces color bias and some artifacts, especially in Figs. 9 and 10. Model ℓ_1-ROF also removes Gaussian noise well, but this method brings about the well-known staircasing artifacts or the oversmooth problem, as shown in Figs. 7(d), 8(d), 9(d), and 10(d). Comparing with these methods, the PGPD model overcomes the problems of color bias and different artifacts with a slight loss of details. As an improvement of PGPD, the proposed pQSTV method avoids the oversmoothness and achieves the best visual quality among all competing methods.

5. Conclusion

In this paper, we proposed a novel color image denoising model which combines the total variation and dictionary learning method for color image denoising. Specially, we proposed a pure quaternion strategy to describe the correlation between channels of color images very well. Secondly, we proposed a novel q-TV regularizer and combined the q-TV with the proposed pQS model. In this way, our method can eliminate artifacts and better preserve the true color of color images, simultaneously. Our model can process three color channels holistically and preserve the correlations of RGB channels. Extensive experiments have demonstrated the effectiveness of our pQSTV method in color image denoising. In the future, we plan to extend this denoising model to other color image processing tasks, like deblurring, inpainting, and super-resolution, etc.

Appendix

Proof of Proposition 1

Proof. It is obvious that both ℓ_1 and ℓ_2 norms are continuous and convex on the Banach Space, so the Eq. (28) is continuous and convex. Additionally for any sequence $\|\dot{X}^k\| \to +\infty$, $\sum_{\iota=0}^{3}\|\mathbf{X}_\iota - \mathbf{Y}_\iota\|_2^2$ will go to infinity. $\sum_{\iota=0}^{3}\|\nabla_\iota \mathbf{X}_\iota\|_1$, $\sum_{ij}\sum_{\iota=0}^{3}\|\mathbf{D}_\iota \mathbf{a}_{\iota_{ij}} - \mathcal{R}_{\iota_{ij}}\mathbf{X}_\iota\|_2^2$, and $\Phi_0(\dot{\mathbf{X}})$ are non-negative. So $J(\mathbf{X})$ will go to infinity, which means $J(\mathbf{X})$ is coercive and has a global minimizer.
Moreover, suppose that $\dot{\mathbf{u}} = \mathbf{u}_0 + \mathbf{u}_1\mathbf{i} + \mathbf{u}_2\mathbf{j} + \mathbf{u}_3\mathbf{k} \in \mathbb{H}^{m\times n}$ and $\dot{\mathbf{v}} = \mathbf{v}_0 + \mathbf{v}_1\mathbf{i} + \mathbf{v}_2\mathbf{j} + \mathbf{v}_3\mathbf{k} \in \mathbb{H}^{m\times n}$ are both minimizers of $J(\mathbf{X})$. Since $J(\mathbf{X})$ is strictly convex, we have for any $0 \le t \le 1$ that

$$J(t\dot{\mathbf{u}} + (1-t)\dot{\mathbf{v}}) = tJ(\dot{\mathbf{u}}) + (1-t)J(\dot{\mathbf{v}}). \tag{37}$$

Since each term of $J(\mathbf{X})$ is convex, then (37) implies that $\dot{\mathbf{u}} = \dot{\mathbf{v}}$. Hence $J(\mathbf{X})$ has a unique minimizer. □

Acknowledgments

The research was supported in part by the Natural Science Foundation of China (Grant No. 61971234, 12126340, 12126304, 11501301, 11771005, 12171210), the "QingLan" Project for Colleges and Universities of Jiangsu Province, the NSF grant 11771188, the Hong Kong Research Grant Council GRF grant 12300218, 12300519, 17201020, 17300021, C1013-21GF, C7004-21GF and Joint NSFC-RGC N-HKU76921, the Natural Science Research of Jiangsu Higher Education Institutions of China under grant 21KJA110001.

References

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[2] Marino Badiale and Enrico Serra. *Semilinear elliptic equations for beginners: existence results via the variational approach.* Springer Science & Business Media, 2010.

[3] Antoni Buades, Bartomeu Coll, and JeanMichel Morel. Image and movie denoising by nonlocal means. *International Journal of Computer Vision*, 76(2):1–25, 2006.

[4] Shuting Cai, Zhao Kang, Ming Yang, Xiaoming Xiong, Chong Peng, and Mingqing Xiao. Image denoising via improved dictionary learning with global structure and local similarity preservations. *Symmetry*, 10(50):167–187, 2018.

[5] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145, 2011.

[6] Yongyong Chen, Xiaolin Xiao, and Yicong Zhou. Low-rank quaternion approximation for color image processing. *IEEE Transactions on Image Processing*, 29:1426–1439, 2020.

[7] Mujibur Rahman Chowdhury, Jun Zhang, Jing Qin, and Yifei Lou. Poisson image denoising based on fractional-order total variation. *Inverse Problems and Imaging*, 14(1):77–96, 2020.

[8] Bartomeu Coll, Joan Duran, and Catalina Sbert. Half-linear regularization for nonconvex image restoration models. *Inverse Problems and Imaging*, 9(2):337–370, 2015.

[9] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen O. Egiazarian. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. *2007 IEEE International Conference on Image Processing*, 1:I313–I316, 2007.

[10] Cássio Fraga Dantas, Jeremy E. J. Cohen, and Rémi Gribonval. Hyperspectral image denoising using dictionary learning. *2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–5, 2019.

[11] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22:1620–1630, 2013.

[12] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing*, 20(7):1838–1857, 2011.

[13] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[14] Kjersti Engan, Sven Ole Aase, and John Håkon Husøy. Multi-frame compression: theory and design. *Signal Processing*, 80:2121–2140, 2000.

[15] Faming Fang, Juncheng Li, Yiting Yuan, Tieyong Zeng, and Guixu Zhang. Multilevel edge features guided network for image denoising. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9):3956–3970, 2020.

[16] Bhawna Goyal, Ayush Dogra, Sunil Agrawal, Balwinder Singh Sohi, and Apoorav Maulik Sharma. Image denoising review: From classical to state-of-the-art approaches. *Information Fusion*, 55:220–244, 2020.

[17] William Rowan Hamilton. *Elements of quaternions.* Longmans, Green, & Company, 1866.

[18] Chaoyan Huang, Zhi Li, Yubing Liu, Tingting Wu, and Tieyong Zeng. Quaternion-based weighted nuclear norm minimization for color image restoration. *Pattern Recognition*, 128:108665, 2022.

[19] Chaoyan Huang, Michael K. Ng, Tingting Wu, and Tieyong Zeng. Quaternion-based dictionary learning and saturation-value total variation regularization for color image restoration. *IEEE Transactions on Multimedia*, pages 1–13, 2021.

[20] Zhigang Jia, Michael K Ng, and Wei Wang. Color image restoration by saturation-value total variation. *SIAM Journal on Imaging Sciences*, 12(2):972–1000, 2019.

[21] Le Jiang, Jun Huang, XiaoGuang Lv, and Jun Liu. Alternating direction method for the high-order total variation-based poisson noise removal problem. *Numerical Algorithms*, 69:495–516, 2014.

[22] Kenneth Kreutz-Delgado, Joseph F. Murray, Bhaskar D. Rao, Kjersti Engan, Te-Won Lee, and Terrence J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15:349–396, 2003.

[23] Juncheng Li, Hanhui Yang, Qiaosi Yi, Faming Fang, Guangwei Gao, Tieyong Zeng, and Guixu Zhang. Multiple degradation and reconstruction network for single image denoising via knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2022.

[24] Xiaoyao Li, Yicong Zhou, and Jing Zhang. Quaternion non-local total variation for color image denoising. *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 1602–1607, 2019.

[25] Chaoyu Liu, Zhonghua Qiao, and Qian Zhang. An active contour model with local variance force term and its efficient minimization solver for multi-phase image segmentation. *arXiv preprint arXiv:2203.09036*, 2022.

[26] Chaoyu Liu, Zhonghua Qiao, and Qian Zhang. Two-phase segmentation for intensity inhomogeneous images by the Allen–Cahn local binary fitting model. *SIAM Journal on Scientific Computing*, 44(1):B177–B196, 2022.

[27] Hao Liu, Xue-Cheng Tai, Ron Kimmel, and Roland Glowinski. A color Elastica model for vector-valued image regularization. *SIAM Journal on Imaging Sciences*, 14(2):717–748, 2021.

[28] Zexin Liu and Wanggen Wan. Image inpainting algorithm based on ksvd and improved cdd. *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, pages 413–417, 2018.

[29] Shousheng Luo, Qian Lv, Heshan Chen, and Jinping Song. Second-order total variation and primal-dual algorithm for CT image reconstruction. *International Journal of Numerical Analysis and Modeling*, 14(1):76–87, 2017.

[30] Liyan Ma, Lionel Moisan, Jian Yu, and Tieyong Zeng. A stable method solving the total variation dictionary model with l^∞ constraints. *Inverse Problems and Imaging*, 8(2):507–535, 2014.

[31] Julien Mairal, Francis R. Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696, 2009.

[32] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17:53–69, 2008.

[33] Michael K. Ng, Xiaoming Yuan, and Wenxing Zhang. Coupled variational image decomposition and restoration model for blurred cartoon-plus-texture images with missing pixels. *IEEE Transactions on Image Processing*, 22:2233–2246, 2013.

[34] ZhiFeng Pang, YaMei Zhou, Tingting Wu, and DingJie Li. Image denoising via a new anisotropic total-variation-based model. *Signal Processing: Image Communication*, 74:140–152, 2019.

[35] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44, 1993.

[36] SooChang Pei and ChingMin Cheng. A novel block truncation coding of color images by using quaternion-moment-preserving principle. *IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World. ISCAS 96*, 2:684–687, 1996.

[37] Zhonghua Qiao and Qian Zhang. Two-phase image segmentation by the Allen-Cahn equation and a nonlocal edge detection operator. *arXiv preprint arXiv:2104.08992*, 2021.

[38] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1–4):259–268, 1992.

[39] Anis Theljani. Non-standard fourth-order PDE related to the image denoising multi-scale non-standard fourth-order PDE in image denoising and its fixed point algorithm. *International Journal of Numerical Analysis and Modeling*, 18:38–61, 2021.

[40] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[41] Ivana Tosic and Pascal Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28:27–38, 2011.

[42] Xiangyang Wang, Qian Wang, Xuebin Wang, HongYing Yang, Zhifang Wu, and PanPan Niu. Color image segmentation using proximal classifier and quaternion radial harmonic fourier moments. *Pattern Analysis and Applications*, pages 1–20, 2019.

[43] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.

[44] Zhengjiang Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004.

[45] Tingting Wu, Zhi-Feng Pang, Youguo Wang, and Yu-Fei Yang. CS-MRI reconstruction based on the constrained TGV-shearlet scheme. *International Journal of Numerical Analysis and Modeling*, 17(3), 2020.

[46] Tingting Wu and Jinbo Shao. Non-convex and convex coupling image segmentation via TGpV regularization and thresholding. *Advances in Applied Mathematics and Mechanics*, 12(3):849–878, 2020.

[47] Tingting Wu, Yichen Zhao, Zhihui Mao, Li Shi, Zhi Li, and Yonghua Zeng. Image segmentation via Fischer-Burmeister total variation and thresholding. *Advances in Applied Mathematics and Mechanics*, 14(4):960–988, 2022.

[48] Jun Xu, Lei Zhang, Wangmeng Zuo, David Zhang, and Xiangchu Feng. Patch group based nonlocal self-similarity prior learning for image denoising. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 244–252, 2015.

[49] Yangyang Xu and Wotao Yin. A fast patch-dictionary method for whole image recovery. *Inverse Problems & Imaging*, 10(2):563, 2016.

[50] Yi Xu, Licheng Yu, Hongteng Xu, Hao Zhang, and Truong Nguyen. Vector sparse representation of color image using quaternion matrix analysis. *IEEE Transactions on Image Processing*, 24(4):1315–1329, 2015.

[51] Fenlin Yang, Ke Chen, and Bo Yu. Efficient homotopy solution and a convex combination of ROF and LLT models for image restoration. *International Journal of Numerical Analysis and Modeling*, 9(4):907927, 2012.

[52] Junfeng Yang, Yin Zhang, and Wotao Yin. An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise. *SIAM Journal on Scientific Computing*, 31:2842–2865, 2009.

[53] Xue Yang and Yu-Mei Huang. A modulus iteration method for SPSD linear complementarity problem arising in image retinex. *Advances in Applied Mathematics and Mechanics*, 12:579–598, 2020.

[54] Yibin Yu, Yulan Zhang, and Shifang Yuan. Quaternion-based weighted nuclear norm minimization for color image denoising. *Neurocomputing*, 332:283–297, 2019.

[55] Fuzhen Zhang. Quaternions and matrices of quaternions. *Linear Algebra and Its Applications*, 251:21–57, 1997.

[56] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

School of Science, Nanjing University of Posts and Telecommunications, Nanjing
E-mail: `wutt@njupt.edu.cn, 1019081805@njupt.edu.cn, jinzhm@njupt.edu.cn`

School of Mathematics and Statistics and Jiangsu Key Laboratory of Education Big Data Science and Engineering, Jiangsu Normal University, Xuzhou
E-mail: `zhgjia@jsnu.edu.cn`

Department of Mathematics, The University of Hong Kong, Hong Kong
E-mail: `mng@maths.hku.hk`