Chinese Journal of Electronics
Vol.32, No.3, May 2023

Quaternion Quasi-Chebyshev Non-local Means for Color Image Denoising

XU Xudong[1], ZHANG Zhihua[1], and M. James C. Crabbe[2]

(1. *School of Mathematics, Shandong University, Jinan 250100, China*)

(2. *Wolfson College, University of Oxford, Oxford OX1 2JD, UK*)

Abstract — Quaternion non-local means (QNLM) denoising algorithm makes full use of high degree self-similarities inside images to suppress the noise, so the similarity metric plays a key role in its denoising performance. In this study, two improvements have been made for the QNLM: 1) For low level noise, the use of quaternion quasi-Chebyshev distance is proposed to measure the similarity of image patches and it has been used to replace the Euclidean distance in the QNLM algorithm. Since the quasi-Chebyshev distance measures the maximal distance in all color channels, the similarity of color images measured by quasi-Chebyshev distance can capture the structural similarity uniformly for each color channel; 2) For high level noise, quaternion bilateral filtering has been proposed as the preprocessing step in the QNLM algorithm. Denoising simulations were performed on 110 images of landscape, people, and architecture at different noise levels. Compared with QNLM, quaternion non-local total variation (QNLTV), and non-local means (NLM) variants (NLTV, NLM after wavelet threshold preprocessing, and the color adaptation of NLM), our novel algorithm not only improved PSNR/SSIM (peak signal to noise rate/structural similarity) and figure of merit values by an average of 2.77 dB/8.96% and 0.0491 respectively, but also reduced processing time.

Key words — Color image denoising, Quaternion quasi-Chebyshev non-local means, Quaternion bilateral filter, Quaternion non-local means.

I. Introduction

Noises are often modeled as white Gaussian noise with zero mean and constant variance. Gray images are often contaminated by various noises during the process of generation, transportation, and processing, leading to serious destruction of the visual effect of images [1]. Denoising is an indispensable step before the images are further subjected to edge detection, feature extraction, and object recognition. Classic filters (e.g., mean filter, median filter, Wiener filter, Gaussian filter, and bilateral filter) are the main tools in the early stages of image denoising [2]–[6], but they tend to localize the processing and ignore the local similarity of the image, resulting in blurred edges and disruption of the main geometry in denoised images.

In contrast to classic denoising filters, the non-local means (NLM) denoising algorithm [7] adopts a novel approach by the use of high degree self-similarities inside images, i.e., many similar image configurations exist in the same gray image. The NLM denoising algorithm does not directly operate pixels as classical filters but operates image patches, so it has good robustness. The NLM denoising has proven to be asymptotically optimal under a generic statistical image denoising algorithm. Since the birth of the NLM algorithm, various improvements have been proposed: the non-local total variation (NLTV) improved the similarity metric of non-local means by matching of the contribution of distantly related pixels to the current pixel [8]. The non-local means-graphics processing unit (NLM-GPU) tried to solve the computational intensity and reduced the runtime of the NLM algorithm through the use of moving average filters for fast Euclidean distance calculation [9]. The non-local means hidden Markov model (NLM-HMM) increased the number of image patches via an HMM-based invariant similarity measure [10]. The non-local means after wavelet threshold preprocessing (WNLM) compensated for the blurring and loss of edge detail in the denoised images of the NLM algorithm by using wavelet thresholding in the high frequency part of the wavelet domain of the image and

Manuscript Received Mar. 17, 2022; Accepted Jan. 7, 2023. This work was supported by European Commission Horizon 2020's Flagship Project (861584) and Taishan Distinguished Professor Fund.
 DOI:10.23919/cje.2022.00.138

NLM in the low frequency part for denoising [11].

Color images contain better visual effects than gray images in terms of visual perception, and the edge information of color images is more abundant than that of gray images, so effectively denoising color images become more difficult than gray images. Denoising algorithms on gray images can be naturally extended to denoise color images by denoising each of the three channels of any RGB color image and then synthesizing the processed channels to produce a new color image [12]. This simple approach ignores internal similarity among the three channels of the color image, possibly leading to produce locally inconsistent colors and then destroying the details of hue, edge and so on. There are two feasible approaches to solving this problem: Buades *et al.* [7] and Goossens *et al.* [9] defined a correlation function to enhance the robustness of weights by calculating the similarity between different channels and proposed the color adaptation of non-local means (NLMC). Another approach is to use the pure quaternion representation to describe the relationships among three channels of RGB color images [13]–[15]. The quaternion non-local means (QNLM) denoising algorithm was developed to apply the NLM denoising algorithm to pure quaternion representations of color images, maintaining denoising consistency between color channels [16], [17].

In this research, the quaternion quasi-Chebyshev non-local means (QCNLM) is proposed to remove low-level noise in RGB color images by incorporating the quaternion quasi-Chebyshev distance into the QNLM algorithm. Compared with the Euclidean distance in the QNLM algorithm, the quasi-Chebyshev distance can better measure the similarity of noisy image patches uniformly for each color channel. When dealing with the removal of the high-level noise ($\sigma_n > 50$), a quaternion bilateral filtering (QBF) is proposed, and the QBF has been used as the preprocessing step of the QCNLM algorithm. Compared with a traditional Gaussian low-pass filter, our QBF not only exploits the spatial proximity of pixels and the similarity of grey values between pixels, but also the relationship between different channels of color images. The denoising simulation of 110 test images under various noise levels verified the significant improvement of our QCNLM to the original QNLM, NLM, and its variation versions (WNLM, NLMC, NLTV, QNLTV) in terms of color peak signal-to-noise ratio and perceived quality.

II. Quaternion Non-local Means Denoising Algorithm

Buades *et al.* [7], [18], [19] proposed an NLM algorithm for gray image denoising. Since the image information always has certain repeatability (self-similarity patterns) while the noise distribution is random, the core idea of NLM is to make use of self-similarity patterns to suppress the noise. Since the NLM algorithm enhances the denoising process from pixel level to patch level, its denoising performance is better than many known denoising algorithms, such as the Gaussian filter, total variation, anisotropic filter, and empirical Wiener filter [20].

The noisy gray image is modeled as $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{N}$. The denoised image $\widehat{\boldsymbol{X}}$ by the NLM algorithm is calculated as follows:

$$\widehat{\boldsymbol{X}}(p) = \frac{\sum\limits_{q \in \boldsymbol{S}_{\rm p}} w(p,q) \times \boldsymbol{Y}(q)}{\sum\limits_{q \in \boldsymbol{S}_{\rm p}} w(p,q)} \tag{1}$$

where $\boldsymbol{S}_p$ is the search window with center p, the weight $w(p,q)$ is

$$w(p,q) = \exp\left(-\frac{d(p,q)/{\sigma_n}^2}{h^2}\right) \tag{2}$$

and $d(p,q)$ represents the Euclidean distance between two image patches with center p and q in the search window $\boldsymbol{S}_p$.

For color image denoising, if the NLM denoising algorithm is used to directly deal with each color channel independently, since the relation among channels is ignored, the denoising effect is often unsatisfied. A pure quaternion representation of RGB channels of color images is proposed as follows:

$$\mathfrak{r}(m,n) = \mathrm{r}(m,n)\mathrm{i} + \mathrm{g}(m,n)\mathrm{j} + \mathrm{b}(m,n)\mathrm{k} \tag{3}$$

where the three imaginary parts $\mathrm{r}(m,n)$, $\mathrm{g}(m,n)$, and $\mathrm{b}(m,n)$ represent Red, Green and Blue channels, respectively, and i, j, and k are three imaginary units [21].

The natural coupling between color channels can be achieved through a pure quaternion representation of the color image. Incidentally, the classic NLM for denoising gray images can be naturally extended to the quaternion non-local means (QNLM) algorithm for denoising color images [22]–[24]. The QNLM denoising algorithm not only has a good denoising effect but also better protects the image edge information. However, when the noise level is above 50, the damage degree of the color image is too large. If the QNLM is used directly, the denoised color images would be blurred and the edge information would be lost. A Gaussian low-pass filter (LPF) is widely suggested to preprocess the noisy images before a QNLM algorithm is applied [22]–[24].

III. Proposed Algorithm

The quaternion non-local means denoising algorithm makes full use of high degree self-similarities inside images to suppress the noise, so the similarity metric among image patches plays a key role in its denoising performance. We develop a novel similarity metric among image patches and replace Euclidean distance in the QNLM algorithm.

For any two color-image patches $\mathfrak{z}_i(m,n)$ and $\mathfrak{z}_j(m,n)$ with size $w \times w$, denote their pure quaternion representation by

$$\mathfrak{C}_i(m,n) = \mathfrak{z}_i^{\mathrm{R}}(m,n)\mathrm{i} + \mathfrak{z}_i^{\mathrm{B}}(m,n)\mathrm{j} + \mathfrak{z}_i^{\mathrm{G}}(m,n)\mathrm{k}$$

$$\mathfrak{C}_j(m,n) = \mathfrak{z}_j^{\mathrm{R}}(m,n)\mathrm{i} + \mathfrak{z}_j^{\mathrm{B}}(m,n)\mathrm{j} + \mathfrak{z}_j^{\mathrm{G}}(m,n)\mathrm{k}$$

where the superscripts R, G and B represent three channels of RGB color images, respectively. The quaternion quasi-Chebyshev distance of image patches $\mathfrak{z}_i$ and $\mathfrak{z}_j$ is defined as

$$\begin{aligned} cd\left(\mathfrak{C}_i, \mathfrak{C}_j\right) &= \max_{m,n} \left|\mathfrak{C}_i(m,n) - \mathfrak{C}_j(m,n)\right| \\ &= \max_{m,n} \left\{ \sqrt{\left[\mathfrak{z}_i^{\mathrm{R}}(m,n) - \mathfrak{z}_j^{\mathrm{R}}(m,n)\right]^2 + \left[\mathfrak{z}_i^{\mathrm{G}}(m,n) - \mathfrak{z}_j^{\mathrm{G}}(m,n)\right]^2 + \left[\mathfrak{z}_i^{\mathrm{B}}(m,n) - \mathfrak{z}_j^{\mathrm{B}}(m,n)\right]^2} \right\} \end{aligned} \tag{4}$$

A quaternion quasi-Chebyshev distance can have obvious advantages over a quaternion Euclidean distance in original QNLM: The quaternion quasi-Chebyshev distance measures the maximal distance in all color channels while the Euclidean distance measures the total distance, so the similarity of color images measured by quasi-Chebyshev distance can capture the structural similarity uniformly for each channel of color images, and that by the Euclidean distance cannot achieve this aim. We use a quaternion quasi-Chebyshev distance to replace quaternion Euclidean distance in the QNLM algorithm and propose a new algorithm quaternion quasi-Chebyshev non-local means (QCNLM) for color image denoising. By coupling quaternion quasi-Chebyshev distance into the classic QNLM algorithm, we propose the QCNLM denoising algorithm as Algorithm 1.

Algorithm 1 Quaternion quasi-Chebyshev non-local means

Input: Noisy image $\mathfrak{Y}$, search window size $W \times W$, patch size $w \times w$, total patch number I, denoising parameter h.

Output: Denoised image $\hat{\mathfrak{X}}$.

for $i = 1 : I$ **do**
  for each patch $\mathfrak{y}_i$ in $\mathfrak{Y}$ **do**
    for each image patch $\mathfrak{y}_j$ in the search window **do**
      Calculate quasi-Chebyshev distance $\mathrm{cd}\left(\mathfrak{C}_i, \mathfrak{C}_j\right)$;
      Calculate normalized parameter
      $N_i = \sum_{j \in I} \exp\left(-\frac{\mathrm{cd}\left(\mathfrak{C}_i, \mathfrak{C}_j\right)/\sigma^2}{h^2}\right)$
      Calculate weight vector
      $\mathrm{cw}_{ij} = \exp\left(-\frac{\mathrm{cd}\left(\mathfrak{C}_i, \mathfrak{C}_j\right)/\sigma^2}{h^2}\right)$
    end for
    Calculate clear image patch $\widehat{\mathfrak{y}_i} = \frac{1}{N_i} \sum_{j \in I} \mathrm{cw}_{ij} \times \mathfrak{y}_j$;
  end for
  Calculate the average of $\mathfrak{y}_i$ and get the estimated clear image window $\widehat{\mathfrak{x}_i}$;
end for
Aggregate all patches together, and get clear image $\hat{\mathfrak{X}}$.
return $\hat{\mathfrak{X}}$

Similar to a classic QNLM algorithm, our QCNLM cannot directly remove the high-level noise level. Different from LPF used as the prepossessing step in the classic QNLM algorithm, we generalize a bilateral filter [6] into the quaternion bilateral filter (QBF) and use the QBF as the prepossessing step in our QCNLM algorithm. The QBF of a pure quaternion representation $\{\mathfrak{F}(u,v)\}_{u,w}$ of any color image is defined as

$$\widehat{\mathfrak{F}}(m,n) = \frac{\sum\limits_{u,v} \mathrm{e}^{-d^2[(m,n),(u,v)]/2\sigma_d^2} \times \mathrm{e}^{-\mathrm{qd}^2[(\mathfrak{F}(m,n),\mathfrak{F}(u,v))]/2\sigma_r^2} \times \mathfrak{F}(u,v)}{\sum\limits_{u,v} \mathrm{e}^{-d^2[(m,n),(u,v)]/2\sigma_d^2} \times \mathrm{e}^{-\mathrm{qd}^2[(\mathfrak{F}(m,n),\mathfrak{F}(u,v))]/2\sigma_r^2}} \tag{5}$$

where $d[\cdot]$ and $\mathrm{qd}[\cdot]$ represent the Euclidean distance and the quaternion Euclidean distance, respectively. The QBF can process three color channels as a whole by representing a color image pixel as a quaternion, ensuring the natural coupling between channels. The QBF makes full use of not only the spatial proximity of pixels and the similarity of gray values between pixels, but also the relation between different channels of color images, so the QBF is better than LPF used in original QNLM. For high level noise, our approach uses QBF first and then followed by QCNLM to denoise color images.

The complete architecture of QCNLM is shown in Fig.1. Except the input and output step, it can be mainly divided into three steps:

i) Preprocessing: if the noise level in a noisy image is greater than 50, the quaternion bilateral filtering is used for pre-processing, otherwise it goes directly to the next step.

ii) Image patch similarity: the noisy image is divided into multiple reference image patches. The qua-

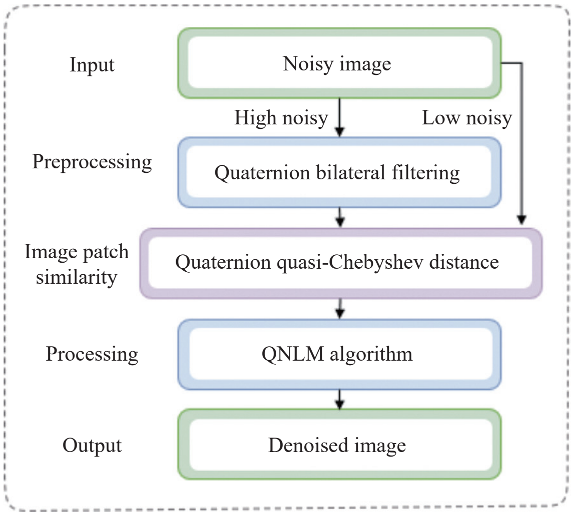

Fig. 1. Patch diagram of the proposed color image denoising algorithm.

ternion quasi-Chebyshev distances between the reference image patches and all the image patches in the window centered on them are calculated as the similarity measure between image patch.

iii) Processing: The similarity measure of image patches is used as the weight in the QNLM denoising algorithm to denoise the image.

IV. Experiments

We demonstrated our proposed QCNLM denoising algorithm's capabilities by comparing it with six denoising algorithms, including three multichannel algorithms (NLM [7], [18], [19], NLTV [8], and WNLM [11]), a channel fusion algorithm (NLMC [9]), and two quaternion-based algorithms (QNLM [22]–[24] and QNLTV [25]). We used 110 test images in total from three datasets: 24 images from Kodak24 [26] (see Fig.2), 68 images from CBSD68 [27] (see Fig.3), and 18 images from McMaster [28] (see Fig.4), and added Gaussian noise at the level $\sigma_n = 10, 15, 25, 35, 55, 75$ to all test images respectively.

The quality of denoising in this paper was measured by the following three indices [29], [30]:

- Peak signal to noise rate (PSNR):

$$\mathrm{PSNR} = 10\log_{10}\frac{\left(2^8-1\right)^2}{\dfrac{1}{M\times N}\displaystyle\sum_{i=1}^{M}\sum_{j=1}^{N}[\mathfrak{X}(i,j)-\widehat{\mathfrak{X}}(i,j)]^2} \tag{6}$$

where $\mathfrak{X}, \widehat{\mathfrak{X}}$ represent original image and denoised image, respectively.

Fig. 2. Kodak24 dataset (enumerated from left-to-right and top-to-bottom).

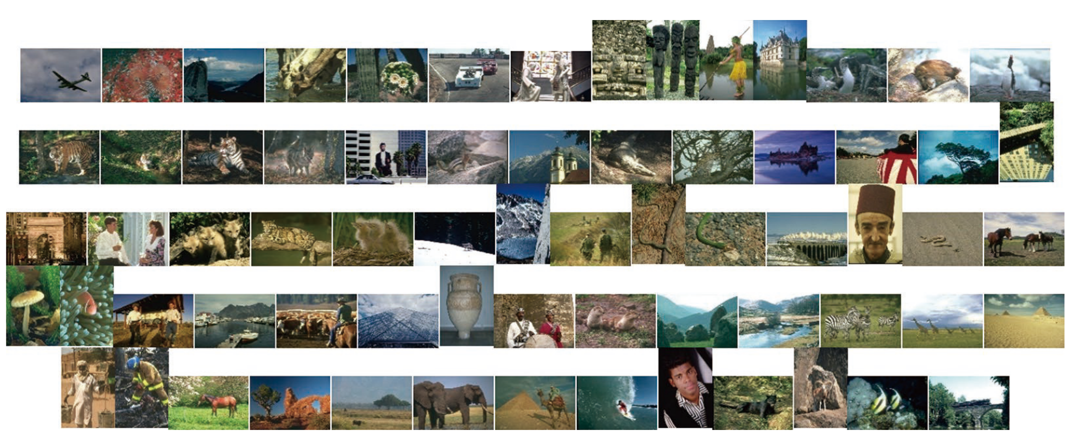

Fig. 3. CBSD68 dataset (enumerated from left-to-right and top-to-bottom).

Fig. 4. McMaster dataset (enumerated from left-to-right and top-to-bottom).

- Structural similarity (SSIM):

$$\begin{aligned}&\mathrm{SSIM}(\mathfrak{X},\hat{\mathfrak{X}})\\&=\frac{1}{N}\sum_{i=1}^{N}\mathrm{SSIM}\left(\mathfrak{x}_i,\hat{\mathfrak{x}}_i\right)\\&=\frac{1}{N}\sum_{i=1}^{N}\frac{(2\mu_{\mathfrak{r}_i}\mu_{\mathfrak{r}_i}+c_1)\times(2\sigma_{\mathfrak{r}_i\mathfrak{t}_i}+c_2)}{({\mu_{\mathfrak{r}_i}}^2+{\mu_{\hat{\mathfrak{r}}_i}}^2+c_1)\times({\sigma_{\mathfrak{r}_i}}^2+{\sigma_{\hat{\mathfrak{r}}_i}}^2+c_2)}\end{aligned} \tag{7}$$

where $\mathfrak{x}_i, \hat{\mathfrak{x}}_i$ denote the corresponding windows of the noisy image $\mathfrak{X}$ and the original image $\hat{\mathfrak{X}}$ indexed by i, respectively; $\mu_{\mathfrak{x}_i}, \mu_{\hat{\mathfrak{x}}_i}, \sigma^2_{\mathfrak{x}_i}, \sigma^2_{\hat{x}_i}, \sigma_{\mathfrak{x}_i\hat{\mathfrak{x}}_i}$ are the mean, variance, and covariance of $\mathfrak{x}_i, \hat{\mathfrak{x}}_i$; the constants c_1, c_2 are to stabilize the division with weak denominator.

- Figure of merit (FOM):

$$\begin{aligned}&\mathrm{FOM}\left(\mathfrak{G}_t,\mathfrak{D}_c\right)\\&=\frac{1}{\max\left(\left|\mathfrak{G}_t\right|,\left|\mathfrak{D}_c\right|\right)}\times\left(\mathrm{TP}+\sum_{p\in\mathrm{FP}}\frac{1}{1+k\times d^2_{\mathfrak{B}_t}(p)}\right)\end{aligned} \tag{8}$$

where $\mathfrak{G}_t$ denotes the reference contour map corresponding to the denoised image, $\mathfrak{D}_c$ denotes the detection contour map of the original image, true positive points (TP) are common points of $\mathfrak{G}_t$ and $\mathfrak{D}_c$: $\mathrm{TP}=|\mathfrak{G}_t\cap\mathfrak{D}_c|$, false positive points (FP) are spurious detected edges of $\mathfrak{D}_c$: $\mathrm{FP}=|\neg\mathfrak{G}_t\cap\mathfrak{D}_c|$ and the parameter k is scaling parameters. The higher FOM, the better the image details are maintained.

For our proposed QCNLM implementation, denoising parameter h, image sub-patch size w and search window size W will influence the denoising effects. For low-level noise, we used the same parameter settings as those in NLMC algorithm [9] (Tables 1 and 2). For high noise levels ($\sigma_n > 50$), due to using QBF for preprocessing the noisy color image, a smaller search range was used (Tables 1 and 2).

Table 1. Parameters in NLMC [9]

σ_n	W	w	h
$0 < \sigma_n \leq 25$	21×21	3×3	0.55
$25 < \sigma_n \leq 50$	35×35	5×5	0.40
$50 < \sigma_n \leq 100$	35×35	7×7	0.35

Table 2. Parameters in QCNLM

σ_n	W	w	h
$0 < \sigma_n \leq 25$	21×21	3×3	0.55
$25 < \sigma_n \leq 50$	35×35	5×5	0.40
$50 < \sigma_n \leq 100$	21×21	5×5	0.40

1. Denoising experiments on the Kodak24 dataset

By using the Kodak24 dataset (Fig.2) with resolution 256×256, we compared our proposed QCNLM algorithm with NLM, NLTV, WNLM, NLMC, QNLM and QNLTV denoising in terms of quantitative, visual and FOM metrics.

1) Low noise level results and comparison

Compared with NLM, NLTV, WNLM, NLMC, QNLM and QNLTV algorithms (Table 3), at the noise level $\sigma_h = 10$, the average PSNR gains of the QCNLM algorithm were 5.95 dB, 4.44 dB, 2.75 dB, 1.98 dB, 1.09 dB and 1.44 dB, and the average SSIM gains were 8.07%, 1.57%, 6.82%, 4.47%, 2.21% and 1.27%; at the noise level $\sigma_n = 15$, the average PSNR gains of the QCNLM algorithm were 5.13 dB, 2.74 dB, 3.52 dB, 2.44 dB, 1.74 dB and 2.58 dB, and the average SSIM gains were 13.26%, 7.97%, 9.98%, 7.48%, 5.08% and 4.27%. Therefore, the QCNLM demonstrated the best denoising performance.

The requirement for a good denoising algorithm is not only to perform denoising well, but also to preserve the details in the image. We used FOM index to evaluate the performance of seven denoising algorithms in retaining details (Table 4). Compared with NLM, NLTV, WNLM, NLMC, QNLM, and QNLTV algorithms, the average FOM gains of the QCNLM algorithm were 0.0447, 0.0409, 0.0428, 0.0355, 0.0109, and 0.0069, respectively (Table 4). This is because the image similarity measured based on quaternion quasi-Chebyshev distances in QCNLM better maintained color consistency than those based on traditional Euclidean distances in QNLM.

For the 9th image in the Kodak24 image dataset destroyed by the $\sigma_n = 10$ Gaussian noise, Figs.5(c)–(i) are the denoised images by the NLM, NLTV, WNLM, NLMC, QNLM, QNLTV and QCNLM algorithms, where the sky gradually became clearer and the color stratification of the sailboat gradually became apparent. When the 9th image was zoomed 5 times (Fig.5), some specific features of the image were revealed to be en-

Table 3. Denoising results at low noise levels (PSNR (dB)/SSIM (%)) using seven algorithms (The best result is in bold)

$\sigma_n = 10$							
Image	**NLM**	**NLTV**	**WNLM**	**NLMC**	**QNLM**	**QNLTV**	**QCNLM**
1	26.94/89.06	23.79/90.59	27.80/83.58	29.31/90.48	29.54/90.88	28.51/85.79	**30.18/91.33**
2	28.62/80.76	30.58/90.01	30.56/83.33	30.60/84.36	31.55/87.07	31.89/90.29	**33.07/90.44**
3	29.00/80.82	31.63/91.84	31.21/86.16	30.30/82.89	32.08/88.36	33.38/**94.58**	**34.03**/92.70
4	28.77/83.46	29.40/90.50	30.40/84.31	30.43/85.63	31.34/87.93	30.28/89.22	**32.82/90.68**
5	20.67/85.00	22.07/87.66	26.04/87.49	28.33/93.02	28.51/93.23	27.21/88.84	**28.81/93.26**
6	25.96/83.88	27.91/88.97	29.18/82.48	29.97/86.98	30.74/88.87	30.74/85.69	**31.61/90.25**
7	27.63/87.50	28.78/93.60	29.23/87.64	30.31/90.29	31.13/92.16	29.86/93.73	**32.12/94.37**
8	23.45/90.60	26.83/92.82	25.53/89.25	28.74/93.50	28.98/93.86	26.53/91.60	**29.12/94.59**
9	27.64/78.90	28.08/91.57	30.89/86.71	29.65/81.51	32.28/88.02	30.48/**96.56**	**34.37**/93.20
10	29.42/81.74	29.31/91.80	30.98/84.45	30.89/85.30	32.08/88.45	29.87/**93.54**	**34.02**/92.32
11	24.09/82.41	25.91/88.91	28.72/83.91	29.70/87.79	30.41/89.59	29.39/89.07	**31.18/91.29**
12	26.10/80.58	26.87/89.71	30.61/84.04	30.60/85.08	31.66/87.84	29.48/90.82	**33.27/91.01**
13	25.23/88.97	26.16/84.60	26.14/82.71	27.80/89.26	27.97/**89.54**	27.81/87.14	**28.19**/88.87
14	22.84/85.76	24.57/89.16	27.72/84.83	29.10/89.91	29.62/90.86	29.37/87.75	**30.02/91.38**
15	27.47/83.36	29.62/90.68	30.26/85.48	29.77/85.04	31.30/88.73	32.21/90.68	**32.56/91.09**
16	29.09/83.02	32.21/90.24	30.38/82.18	30.51/85.44	31.52/87.87	32.92/87.72	**32.94/90.27**
17	24.17/83.41	26.15/90.76	28.93/86.20	30.03/88.28	30.84/90.39	31.06/92.24	**31.93/92.94**
18	24.21/84.98	25.83/90.20	28.52/83.56	29.77/88.19	30.48/89.85	30.73/88.74	**31.32/91.42**
19	25.56/81.73	27.90/90.89	29.66/85.47	30.36/86.49	31.35/89.23	32.04/92.17	**32.58/92.42**
20	24.91/85.67	26.52/92.53	30.43/89.34	31.15/89.70	32.08/91.76	32.38/**94.52**	**33.36**/94.09
21	24.58/81.64	26.56/90.18	28.66/85.39	29.71/86.96	30.47/89.44	30.48/91.86	**31.29/92.18**
22	26.63/83.45	28.54/89.95	29.45/82.87	30.15/86.57	30.99/88.53	31.31/88.72	**32.05/90.51**
23	27.82/81.62	30.42/91.92	31.08/86.73	30.71/85.69	31.82/88.72	32.54/**94.25**	**33.62**/92.68
24	21.19/82.82	22.66/88.05	26.42/83.31	29.35/89.19	29.94/90.61	29.76/88.73	**30.47/91.62**
$\sigma_n = 15$							
Image	**NLM**	**NLTV**	**WNLM**	**NLMC**	**QNLM**	**QNLTV**	**QCNLM**
1	25.39/83.17	26.93/77.74	26.06/79.71	26.38/83.92	27.44/86.45	27.01/86.63	**28.31/87.27**
2	26.01/67.39	28.27/79.44	27.77/73.46	26.97/70.64	28.68/77.99	28.23/79.04	**30.67/84.55**
3	26.18/67.35	30.11/86.83	28.15/75.50	27.08/70.88	28.98/79.19	28.53/80.04	**31.33/87.26**
4	25.95/71.21	27.68/84.08	27.67/75.92	26.84/73.34	28.51/79.68	27.48/80.72	**30.47/85.29**
5	21.82/84.63	25.78/83.12	23.40/83.64	25.90/88.75	26.70/90.23	26.27/90.22	**27.27/90.58**
6	24.99/74.27	29.16/78.62	26.71/75.53	26.73/76.63	28.14/81.48	28.09/82.09	**29.55/84.87**
7	25.77/78.99	28.18/90.13	26.90/81.16	26.85/81.21	28.29/86.11	27.43/86.63	**29.78/90.49**
8	24.62/87.78	25.26/87.62	21.75/83.61	26.15/88.70	27.08/90.86	25.65/90.95	**27.50/92.04**
9	25.55/64.64	28.43/86.76	27.57/74.66	27.07/68.86	29.06/78.31	27.40/79.09	**33.98/94.18**
10	26.20/68.27	26.95/82.65	28.02/74.41	28.49/77.78	28.92/79.26	27.18/80.32	**31.29/86.75**
11	24.23/74.60	25.15/71.08	26.15/77.13	27.58/81.80	27.37/80.76	27.21/83.54	**29.25/86.57**
12	25.27/68.65	27.32/82.73	27.57/74.69	28.31/77.63	27.96/76.02	26.98/80.21	**30.84/85.49**
13	20.52/78.39	24.83/74.28	23.38/78.46	26.09/84.77	25.82/84.09	25.26/**85.42**	**26.86**/84.93
14	23.36/80.24	27.88/82.00	25.11/79.96	27.03/84.98	27.36/85.57	27.37/86.18	**28.30/87.42**
15	25.79/72.37	28.21/80.57	27.54/77.31	28.12/79.65	28.54/80.95	28.42/81.90	**30.34/86.02**
16	26.08/70.39	29.28/74.31	27.74/73.83	28.15/77.84	28.59/79.05	28.71/80.18	**30.53/84.26**
17	24.15/74.82	25.59/75.28	25.96/78.48	27.86/82.52	28.23/83.55	27.11/83.62	**29.84/88.48**
18	24.09/76.34	25.29/65.86	25.63/77.13	27.62/82.24	27.98/83.16	28.03/83.88	**29.36/86.70**
19	24.88/70.74	25.32/76.06	26.68/76.15	28.07/79.81	28.49/81.20	28.50/82.10	**30.26/87.48**
20	24.78/76.47	28.43/87.44	27.02/82.09	29.01/84.34	29.41/85.43	28.90/86.27	**31.22/90.31**
21	24.30/72.18	25.74/76.17	26.23/77.04	27.59/80.71	27.96/81.89	27.75/82.64	**29.34/87.37**
22	25.25/72.93	26.66/71.34	26.91/75.48	27.88/79.93	28.28/80.85	21.62/78.29	**29.93/85.41**
23	25.91/68.67	28.77/86.46	28.05/76.62	28.36/78.38	28.79/79.86	28.28/80.82	**31.06/87.47**
24	21.77/75.47	24.85/66.34	23.55/76.56	27.25/83.42	27.59/84.21	27.60/84.96	**28.69/87.07**

hanced: the QCNLM algorithm denoised the image (Fig.5(i)) with fewer noise points in the yellow area of the sailboat than the other six denoising algorithms (Figs.5(c)–(h)), and the numbers "14255" in the enlarged frame were clearer and the outline of the sailboat was closer to the original image.

For the 3rd image in the Kodak24 image dataset destroyed by the $\sigma_n = 15$ Gaussian noise, the proposed QCNLM algorithm (Fig.6(i)) preserved the hat color better than the other six algorithms (Figs.6(c)–(h)), especially, the boundary between the clouds and the sky was clear, the shadows on the walls were flat and the

Table 4. Denoising results at low noise levels (FOM) using seven algorithms (The best result is in bold)

	$\sigma_n = 10$							$\sigma_n = 15$						
Image	**NLM**	**NLTV**	**WNLM**	**NLMC**	**QNLM**	**QNLTV**	**QCNLM**	**NLM**	**NLTV**	**WNLM**	**NLMC**	**QNLM**	**QNLTV**	**QCNLM**
1	0.9304	0.9125	0.9277	0.9028	0.9037	0.9260	0.9315	0.8631	0.9187	0.8840	0.9250	0.8882	0.9166	0.9329
2	0.8763	0.8532	0.831	0.8399	0.8918	0.9122	0.9174	0.8081	0.8145	0.8105	0.8035	0.8314	0.8853	0.8778
3	0.9151	0.9216	0.9129	0.9088	0.9189	0.9397	0.9537	0.8816	0.8941	0.8823	0.8978	0.8996	0.9368	0.9345
4	0.9036	0.8690	0.8511	0.8667	0.9133	0.9171	0.9221	0.8164	0.7805	0.8193	0.8446	0.8330	0.8782	0.8929
5	0.9345	0.9553	0.9374	0.9284	0.9473	0.9448	0.9598	0.9321	0.9257	0.9179	0.9342	0.9376	0.9433	0.9464
6	0.8978	0.8833	0.8498	0.8530	0.9081	0.9036	0.9293	0.8137	0.8473	0.8411	0.8727	0.8994	0.8964	0.8945
7	0.9382	0.9066	0.9553	0.9270	0.9379	0.9442	0.9428	0.8912	0.9245	0.9121	0.9219	0.9295	0.9258	0.9392
8	0.9635	0.9068	0.9729	0.9573	0.9671	0.9520	0.9649	0.9457	0.9057	0.9233	0.9538	0.9546	0.9381	0.9741
9	0.9428	0.9496	0.8977	0.9511	0.9352	0.9298	0.9458	0.9173	0.9172	0.9128	0.9222	0.9379	0.9225	0.9336
10	0.9334	0.8769	0.9140	0.9246	0.9433	0.9426	0.9512	0.6987	0.8784	0.866	0.8581	0.9079	0.9124	0.9189
11	0.9332	0.8943	0.9127	0.9044	0.9533	0.9533	0.9599	0.7983	0.8692	0.8946	0.9211	0.9444	0.9442	0.9427
12	0.9227	0.8589	0.8401	0.9376	0.9634	0.9660	0.9679	0.7988	0.8630	0.9218	0.9182	0.9547	0.9483	0.9565
13	0.8881	0.8871	0.9413	0.9574	0.9293	0.9135	0.9476	0.8760	0.8814	0.8741	0.8875	0.9220	0.9200	0.9002
14	0.9121	0.9288	0.9218	0.9088	0.9416	0.9402	0.9409	0.8893	0.9198	0.8904	0.9037	0.9264	0.9221	0.9297
15	0.9215	0.9239	0.9206	0.8999	0.9297	0.9414	0.9441	0.8897	0.8973	0.8915	0.8743	0.9075	0.9192	0.9252
16	0.8801	0.9135	0.8604	0.8609	0.9105	0.9218	0.9088	0.8577	0.8772	0.8171	0.8834	0.9001	0.8980	0.9032
17	0.9297	0.9375	0.9269	0.9084	0.9513	0.9392	0.9550	0.8960	0.8941	0.8940	0.9200	0.9393	0.9318	0.9414
18	0.9271	0.9230	0.9162	0.8773	0.9424	0.9367	0.9442	0.8679	0.9160	0.8740	0.9029	0.9200	0.9262	0.9147
19	0.9105	0.9242	0.9502	0.8803	0.9354	0.9348	0.9393	0.8937	0.9203	0.8781	0.9212	0.9234	0.9299	0.9348
20	0.9287	0.8869	0.8615	0.8933	0.9478	0.9425	0.9489	0.8811	0.8671	0.8853	0.8929	0.9222	0.9326	0.9274
21	0.9262	0.9163	0.9376	0.9372	0.9559	0.9494	0.9545	0.8934	0.9095	0.8906	0.9179	0.9369	0.9395	0.9405
22	0.9207	0.8841	0.8793	0.8654	0.9400	0.9320	0.9387	0.8382	0.8439	0.8540	0.8956	0.9093	0.9157	0.9209
23	0.8893	0.8666	0.8327	0.8553	0.8996	0.9042	0.9115	0.7990	0.7885	0.8397	0.8098	0.8618	0.8658	0.8877
24	0.9057	0.919	0.9342	0.8963	0.9488	0.9261	0.9429	0.9077	0.9137	0.9150	0.9028	0.9093	0.9380	0.9376
Avg.	0.9179	0.9041	0.9035	0.9017	0.9339	0.9338	**0.9426**	0.8606	0.8819	0.8787	0.8952	0.9123	0.9202	**0.9253**

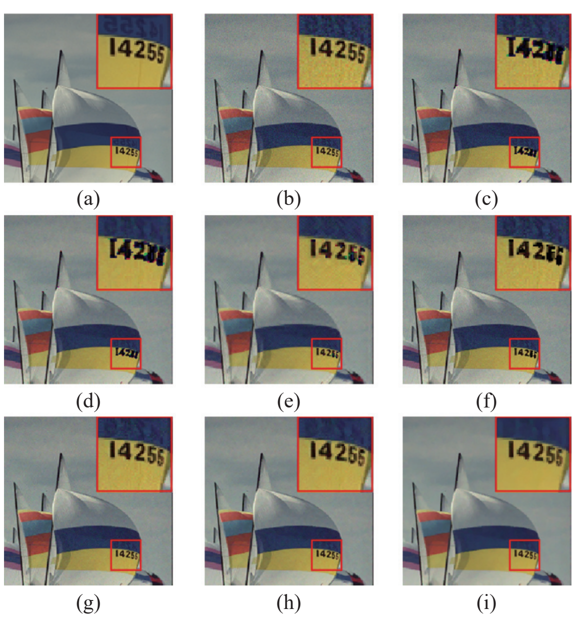

Fig. 5. The visual result image of different denoising algorithms in the 9th image from Kodak24. (a) Noise-free 9th image; (b) The 9th image destroyed by Gaussian noise with $\sigma_n = 10$; (c)–(i) Denoising by NLM, NLTV, WNLM, NLMC, QNLM, QNLTV, and QCNLM, respectively.

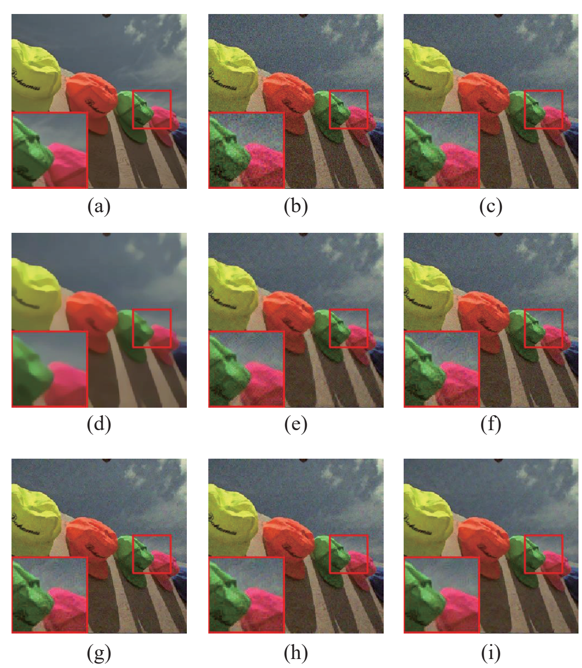

Fig. 6. The visual result image of different denoising algorithms in the 3rd image from Kodak24. (a) Noise-free 3rd image; (b) The 3rd image destroyed by Gaussian noise with $\sigma_n = 15$; (c)–(i) denoising by NLM, NLTV, WNLM, NLMC, QNLM, QNLTV, and QCNLM, respectively.

letters on the hat were relatively clear.

2) Middle noise level results and comparison

Compared with NLM, NLTV, WNLM, NLMC, QNLM and QNLTV algorithms (Table 5), at the noise level $\sigma_n = 25$, the average PSNR gains of the QCNLM algorithm were 4.98 dB, 3.66 dB, 3.61 dB, 2.72 dB, 1.55 dB and 2.44 dB, and the average SSIM gains were 20.65%, 12.49%, 16.04%, 10.7%, 5.77% and 7.59%; at the noise level $\sigma_n = 35$, the average PSNR gains of the QCNLM algorithm were 7.47 dB, 3.51 dB, 5.02 dB, 1.42 dB, 1.34 dB and 1.58 dB, and the average SSIM gains were 31.50%, 11.4%, 24.10%, 4.92%, 5.27% and 1.85%.

In the middle level noise, the average FOM values

Table 5. Denoising results at middle noise levels (PSNR (dB)/SSIM (%)) using seven algorithms (The best result is in bold)

	$\sigma_n = 25$						
Image	**NLM**	**NLTV**	**WNLM**	**NLMC**	**QNLM**	**QNLTV**	**QCNLM**
1	21.70/69.20	23.36/76.40	22.71/68.77	23.28/73.39	23.88/74.97	24.01/76.85	**25.44/77.83**
2	22.07/45.98	24.56/61.46	23.60/52.72	24.02/56.08	24.76/59.88	24.98/63.52	**27.23/71.71**
3	21.99/46.44	24.22/59.87	23.63/54.14	24.06/57.27	24.86/61.36	25.12/64.98	**30.41/86.73**
4	21.89/50.65	23.73/62.72	23.45/56.86	23.82/59.59	24.58/63.10	24.52/66.39	**29.41/81.09**
5	21.62/78.49	22.93/82.74	22.41/78.16	22.11/78.60	23.50/82.26	23.56/83.36	**24.73/84.14**
6	21.84/49.13	23.28/56.74	23.44/54.64	25.23/64.88	26.70/70.64	24.80/69.15	**27.15/71.94**
7	21.82/62.63	23.53/72.15	23.03/66.07	22.42/64.67	24.34/73.00	24.35/75.54	**26.36/80.96**
8	21.65/78.12	22.12/81.18	22.40/78.57	23.92/84.17	24.22/84.72	23.25/84.31	**24.78/86.08**
9	24.32/57.11	28.12/73.18	26.68/70.50	27.13/72.56	29.03/88.39	24.52/63.48	**30.50/88.80**
10	21.88/46.84	22.78/54.92	23.51/53.35	25.45/65.04	26.99/72.04	24.39/64.97	**27.52/74.18**
11	21.86/59.17	23.16/66.68	23.12/62.54	22.91/62.94	24.30/68.78	24.32/71.44	**26.25/76.37**
12	21.97/48.75	23.11/58.37	23.56/55.09	24.46/60.78	27.81/75.95	24.32/65.52	**29.66/82.19**
13	21.45/71.42	22.22/75.85	22.35/71.05	23.09/74.59	23.96/76.43	23.21/74.31	**24.62/77.00**
14	21.72/67.02	19.02/72.13	22.86/68.85	23.64/73.42	24.66/76.51	24.26/76.15	**25.61/78.91**
15	22.15/53.51	23.77/62.30	23.62/59.47	24.45/64.30	25.70/69.93	25.09/68.58	**27.09/75.39**
16	21.92/56.59	23.21/62.90	23.29/60.00	24.91/68.79	26.16/72.92	25.10/65.25	**26.49/73.52**
17	22.08/53.51	23.62/62.30	23.40/59.47	24.64/64.30	26.32/69.93	24.88/72.52	**26.68/75.39**
18	21.99/59.79	23.24/65.88	23.22/62.69	24.45/69.79	25.23/72.51	24.78/71.82	**26.40/76.50**
19	21.86/52.35	23.21/59.52	23.28/57.87	24.64/66.00	25.48/69.93	24.96/68.40	**26.81/76.03**
20	22.99/59.01	24.17/66.56	24.45/65.89	25.74/72.53	26.59/76.08	25.80/74.68	**27.97/81.53**
21	21.74/55.70	22.98/62.23	23.08/60.34	24.32/67.63	25.13/71.07	24.57/69.89	**26.30/76.47**
22	21.83/54.04	23.13/61.08	23.27/58.51	24.54/66.21	25.40/69.33	24.78/68.45	**26.76/74.49**
23	21.94/47.56	23.21/55.79	23.57/55.40	24.88/63.37	25.80/67.90	24.96/65.88	**27.42/75.54**
24	21.61/61.16	22.87/67.23	22.91/63.98	24.10/71.14	24.87/73.66	24.41/73.04	**25.92/77.12**
	$\sigma_n = 35$						
Image	**NLM**	**NLTV**	**WNLM**	**NLMC**	**QNLM**	**QNLTV**	**QCNLM**
1	19.12/56.36	21.87/65.29	20.88/51.28	22.57/63.75	22.70/63.89	24.13/**73.65**	**24.54**/65.68
2	19.67/33.42	24.68/62.96	22.45/44.15	25.76/65.50	25.93/65.95	25.97/70.80	**27.87/74.94**
3	19.38/34.05	24.82/65.13	22.47/47.16	26.07/69.03	26.25/69.26	26.28/74.31	**28.64/80.65**
4	19.32/37.23	21.58/50.25	21.65/46.58	25.58/67.68	25.76/67.99	25.44/72.41	**27.85/75.54**
5	18.38/66.17	21.30/70.14	20.47/68.96	22.67/70.63	22.81/70.86	23.29/**79.93**	**23.35**/73.97
6	19.23/43.04	22.50/58.55	21.46/48.93	24.95/66.62	25.13/66.71	25.46/69.37	**26.50/71.09**
7	19.13/50.40	22.04/65.59	21.13/56.25	24.54/75.23	24.70/75.29	24.71/78.52	**26.16/79.04**
8	17.49/66.13	21.05/77.41	20.36/70.11	21.93/78.90	22.03/78.93	22.81/**82.70**	**23.19**/81.01
9	19.12/31.93	22.40/51.31	21.66/42.78	27.58/77.58	27.74/78.08	25.47/74.36	**29.28/86.54**
10	19.19/33.66	22.30/52.92	21.65/43.29	27.41/74.33	27.57/74.50	25.33/73.39	**28.79/79.09**
11	19.11/46.18	22.12/61.30	21.24/51.95	25.44/73.29	25.59/69.58	24.80/**74.16**	**25.97**/73.34
12	19.28/35.87	22.31/54.05	21.70/45.36	27.09/74.01	25.92/68.00	25.21/73.25	**28.66/79.72**
13	18.45/57.93	20.54/62.27	20.48/59.99	22.80/62.06	22.96/62.21	22.93/72.56	**23.50/63.04**
14	18.79/53.16	20.15/59.19	20.97/58.36	22.63/67.38	22.79/67.47	24.55/70.39	**24.71/76.33**
15	19.92/41.91	25.04/70.80	22.61/52.73	27.52/77.88	27.66/78.01	25.98/74.32	**28.02/78.90**
16	19.44/36.17	25.48/67.04	22.48/45.83	27.80/73.03	27.93/72.83	26.19/70.69	**28.42/73.65**
17	19.73/47.15	24.32/73.23	22.17/55.42	26.59/78.73	26.74/76.70	25.42/76.76	**26.78/78.22**
18	19.68/47.00	23.71/66.93	21.98/52.39	26.14/68.50	25.77/68.53	25.33/**74.36**	**26.24**/70.42
19	19.44/40.96	25.35/73.42	21.95/49.21	25.23/69.81	25.37/70.10	25.56/74.31	**27.48/79.38**
20	20.70/46.60	26.40/80.77	23.31/60.10	26.40/76.94	26.55/77.20	26.62/80.94	**28.72/85.71**
21	19.31/44.43	24.80/74.17	21.77/51.33	24.55/69.98	24.71/70.37	25.14/75.00	**26.23/77.15**
22	19.26/40.49	24.08/65.48	21.92/47.85	24.36/63.98	24.52/63.93	25.58/**72.79**	**26.66**/71.90
23	19.41/34.71	24.36/64.14	22.41/48.56	27.78/78.68	27.92/78.84	26.01/74.46	**28.05/79.13**
24	19.19/47.93	23.56/69.09	21.46/51.83	23.69/67.13	23.86/67.17	24.79/70.98	**25.50/74.45**

of the QCNLM algorithm outperformed the other six competing denoising algorithms (NLM, NLTV, WNLM, NLMC, QNLM and QNLTV) by 0.0897, 0.0825, 0.0542, 0.0509, 0.0203 and 0.0073 at noise level $\sigma_n = 25$, and by 0.1353, 0.0817, 0.0685, 0.0657, 0.0320 and 0.0049 at noise level $\sigma_n = 35$ (Table 6). The QCNLM algorithm could sharpen the images while still performing well in the FOM index for texture and detail metrics.

Table 6. Denoising results at middle noise levels (FOM) using seven algorithms (The best result is in bold)

	$\sigma_n = 25$							$\sigma_n = 35$						
Image	**NLM**	**NLTV**	**WNLM**	**NLMC**	**QNLM**	**QNLTV**	**QCNLM**	**NLM**	**NLTV**	**WNLM**	**NLMC**	**QNLM**	**QNLTV**	**QCNLM**
1	0.9177	0.8896	0.8930	0.8626	0.9124	0.9097	0.8899	0.8384	0.7537	0.8494	0.8306	0.8672	0.8520	0.8468
2	0.6000	0.6507	0.6929	0.7471	0.7459	0.7730	0.8372	0.5149	0.6838	0.6500	0.6996	0.7150	0.7976	0.7946
3	0.7273	0.8026	0.8330	0.7555	0.8309	0.8479	0.8532	0.4724	0.5997	0.6281	0.8136	0.6769	0.8135	0.8250
4	0.6927	0.7451	0.7806	0.7844	0.7788	0.8271	0.8362	0.5191	0.6678	0.5573	0.6887	0.688	0.7745	0.7743
5	0.8936	0.9175	0.9183	0.9066	0.8950	0.9186	0.9360	0.8864	0.8826	0.8905	0.8574	0.9108	0.9053	0.9091
6	0.7502	0.7945	0.8651	0.8403	0.8871	0.8717	0.8873	0.6888	0.7439	0.7700	0.6945	0.7774	0.7720	0.7998
7	0.8658	0.8981	0.9133	0.9119	0.8466	0.9015	0.9226	0.7801	0.8173	0.8257	0.8349	0.8497	0.8986	0.8986
8	0.9096	0.8938	0.8648	0.9182	0.9377	0.9252	0.9707	0.832	0.8659	0.8705	0.9576	0.9273	0.9032	0.9038
9	0.9313	0.8670	0.8645	0.9395	0.9604	0.9115	0.9209	0.7874	0.8349	0.8384	0.8913	0.8792	0.9366	0.9377
10	0.7203	0.6471	0.8282	0.7466	0.8581	0.8735	0.8597	0.5221	0.6582	0.5947	0.6929	0.6772	0.8032	0.8071
11	0.8292	0.8403	0.8309	0.8469	0.8761	0.8979	0.8931	0.7596	0.8184	0.8175	0.7881	0.8414	0.8603	0.8682
12	0.8658	0.7752	0.7978	0.8601	0.9148	0.9257	0.9095	0.7957	0.8364	0.7782	0.865	0.8106	0.9048	0.9118
13	0.8206	0.8036	0.6673	0.8270	0.8660	0.8819	0.9069	0.8282	0.8225	0.8107	0.7648	0.8541	0.8527	0.8552
14	0.8404	0.8553	0.8960	0.8521	0.8947	0.9116	0.9140	0.7865	0.8281	0.8050	0.7971	0.8936	0.8750	0.8752
15	0.8291	0.8596	0.8826	0.8470	0.8848	0.8960	0.8952	0.7438	0.8257	0.8409	0.7930	0.8734	0.8519	0.8636
16	0.6854	0.7027	0.7974	0.8323	0.8388	0.8420	0.8503	0.5369	0.6392	0.6764	0.6334	0.7088	0.6805	0.6965
17	0.8152	0.8529	0.8878	0.8380	0.8887	0.9103	0.9175	0.7798	0.8017	0.8704	0.8002	0.8737	0.8365	0.8577
18	0.8031	0.8330	0.8675	0.8627	0.8707	0.8750	0.8824	0.7651	0.7931	0.8344	0.7108	0.8617	0.8295	0.8305
19	0.7412	0.7615	0.8182	0.8230	0.8408	0.8827	0.8705	0.672	0.7335	0.7994	0.7437	0.7876	0.8288	0.8333
20	0.8895	0.8507	0.8705	0.8801	0.9209	0.9261	0.9290	0.8339	0.8164	0.8621	0.8360	0.8645	0.8943	0.8957
21	0.7981	0.8261	0.8576	0.8581	0.8763	0.8862	0.8856	0.7432	0.7953	0.8653	0.7812	0.8574	0.8661	0.8658
22	0.7857	0.7905	0.8307	0.8332	0.8582	0.8773	0.8766	0.6645	0.7166	0.7662	0.7317	0.8271	0.8088	0.8119
23	0.6959	0.6949	0.7490	0.7454	0.8236	0.8408	0.8409	0.4956	0.6206	0.5957	0.7408	0.6670	0.7834	0.7678
24	0.8376	0.8638	0.8901	0.8550	0.9031	0.9088	0.9127	0.8106	0.7886	0.8655	0.7822	0.8483	0.8587	0.8749
Avg.	0.8018	0.8090	0.8373	0.8406	0.8712	0.8842	**0.8915**	0.7107	0.7643	0.7775	0.7803	0.814	0.8411	**0.8460**

Fig.7 illustrates the visual effect of the 7th image in the Kodak24 dataset corrupted by the $\sigma_n = 25$ noise restored using different algorithms. Compared with NLM, NLTV, WNLM, NLMC, QNLM and QNLTV denoising algorithm (Figs.7(c)–(h)), the QCNLM algorithm (Fig.7(i)) better preserved complex textures and curves, such as the color and background of plants, and windows background details. When the red and pink flowers of the 7th image was zoomed 5 times (Fig.7), the QCNLM algorithm denoised the image (Fig.7(i)) with fewer noise points in the windows than the other six denoising algorithms (Figs.7(c)–(h)). The denoised image by QCNLM demonstrated the best visual experience for the two flowers in terms of color and retained the edges of the leaves on the small red flower best.

For the 15th image in the Kodak24 dataset destroyed by the $\sigma_n = 35$ Gaussian noise, our proposed algorithm QCLNM had the highest PSNR and SSIM among the studied denoising algorithms. The QCNLM not only restored sharp edges and better detail information, but also handled smoothed areas very well with good denoising. The denoised 15th Image by QCLMN had the highest FOM value (0.8498), which means the retaining of the finer details of the original image despite the richness of detail in the test image. Compared to the other six denoising algorithms (Figs.8(c)–(h)), the QCNLM algorithm (Fig.8(i)) enhanced the black and white contrast of the eyes, better restored the wrinkles of the eyelids, and better highlighted the blue-yellow color of the hair band in the black hair.

3) High noise level results and comparison

We compare the classical NLM, NLTV, WNLM, NLMC, QNLM and QNLTV algorithms in LPF preprocessing with our proposed QCNLM algorithms under different preprocessing algorithms of LPF and QBF (LPF+QCNLM and QBF+QCNLM).

At the noise level $\sigma_n = 55$, the PSNR/SSIM values of the QBF+QCNLM were on average 2.0183 dB and 9.44% higher than the other seven algorithms, respectively (Table 7). At the noise level $\sigma_n = 75$, the PSNR/SSIM values of the proposed QBF+QCNLM al-

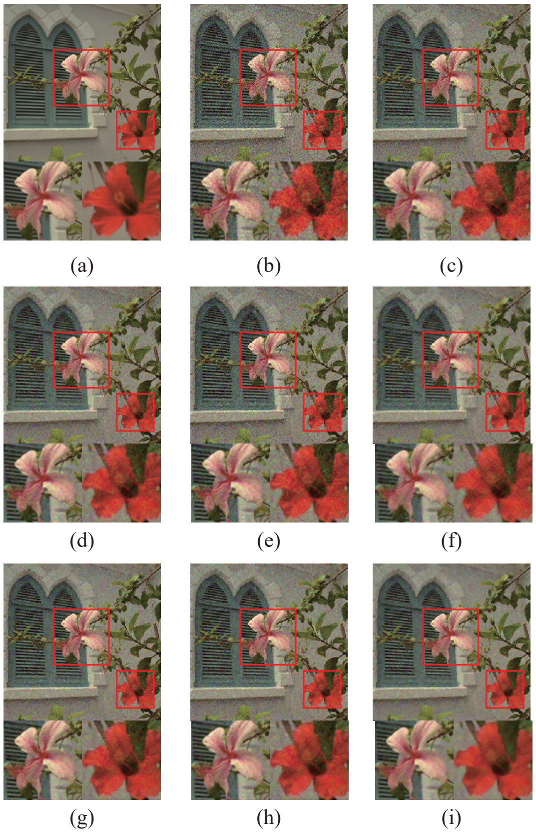

Fig. 7. Visualization of local zooms with different denoising algorithms in 7th image from Kodak24. (a) Noise-free 7th image; (b) The 7th image destroyed by $\sigma_n = 25$ Gaussian noise; (c)–(i) Denoising by NLM, NLTV, WNLM, NLMC, QNLM, QNLTV, and QCNLM, respectively.

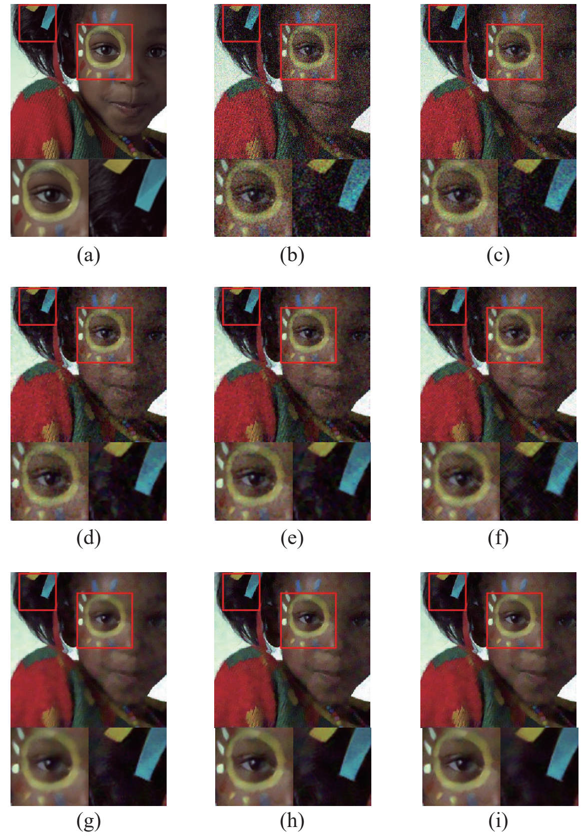

Fig. 8. Visualization of local zooms with different denoising algorithms in 15th image from Kodak24. (a) Noise-free 15th image; (b) The 15th image destroyed by Gaussian noise with $\sigma_n = 35$; (c)–(i) Denoising by NLM, NLTV, WNLM, NLMC, QNLM, QNLTV, and QCNLM, respectively.

gorithm were on average 1.7226 dB and 8.71% higher than the other seven algorithms (Table 7). Since quaternion quasi-Chebyshev distance can measure the similarity between image patches better than Euclidean distance and the QBF considers the spatial proximity of pixels and the similarity of gray values between pixels, and the relation between different channels of color images, our proposed QCNLM denoising algorithm after QBF preprocessing (QBF+QCNLM) was better than NLM, NLTV, WNLM, NLMC, QNLM, QNLTV and LPF+QCNLM algorithms under the high noise level ($\sigma_n > 50$).

The FOM index is a metric used to reflect the ability to preserve detail at the edges of an image. Table 8 shows the FOM values after denoising the exploration dataset at Kodak24 by different algorithms. At high noise levels, the combination of QBF and QCNLM performed better, and the FOM values of QBF+QCNLM algorithm and LPF+QCNLM algorithm outperformed the other six algorithms on average by 0.0573 and 0.0668 ($\sigma_n = 55$) and by 0.0474 and 0.0851 ($\sigma_n = 75$).

For the 24th image in the Kodak24 dataset destroyed by the $\sigma_n = 55$ Gaussian noise, the proposed QCLNM algorithm had the highest FOM value among the denoising algorithms studied, on average 0.01005 higher than the other six algorithms, which means that it better preserved the finer texture information of the original image. Fig.9 shows a contour plot of the 24th image after denoising using the different algorithms. Compared with NLM, NLTV, WNLM, NLMC, QNLM and QNLTV algorithms (Figs.9(c)–(h)), the denoised image generated by LPF+QCNLM algorithm (Fig.9(i)) showed higher visual quality and fewer artifacts, but the image still had a small amount of residual noise that damaged the structure of houses and murals. Since QBF takes full advantage of the spatial proximity of pixels and the similarity of grey values between pixels, the denoising combining the QBF and QCNLM algorithms (Fig.9(j)) performed better in reducing image noise points, preserving the boundary contours of the house mural and recovering the right-hand part of the eaves.

Fig.10 illustrates the denoising performance of the 16th image in the Kodak24 dataset damaged by the

Table 7. Denoising results at high noise levels (PSNR (dB)/SSIM (%)) using eight algorithms (The best result is in bold)

	$\sigma_n = 55$							
Image	**NLM**	**NLTV**	**WNLM**	**NLMC**	**QNLM**	**QNLTV**	**LPF+QCNLM**	**QBF+QCNLM**
1	16.60/41.99	19.39/53.74	20.27/52.38	20.03/52.57	19.70/54.55	20.52/44.47	22.02/53.32	**22.58/54.81**
2	20.43/51.82	21.20/49.77	23.29/52.40	23.38/55.07	25.05/67.04	22.71/50.75	24.88/71.85	**25.08/73.46**
3	20.94/43.29	21.09/45.34	22.39/51.82	21.93/49.77	23.81/63.24	22.84/55.85	23.09/57.46	**25.07/70.32**
4	20.16/41.94	20.85/47.14	21.47/48.58	21.15/47.39	22.99/60.67	22.33/54.23	22.54/54.93	**24.98/67.70**
5	16.19/54.12	16.88/57.68	16.65/54.98	17.02/56.42	**21.42/70.96**	18.28/60.54	18.28/59.91	21.14/**61.11**
6	18.58/41.97	20.86/49.72	21.90/54.94	21.57/54.37	23.21/59.68	22.31/58.23	22.49/58.71	**23.74/60.45**
7	19.79/52.00	20.35/56.29	20.80/55.02	20.70/56.49	22.43/62.35	21.70/61.36	21.85/61.53	**22.96/66.35**
8	17.22/52.66	18.65/50.94	17.25/60.42	19.16/67.21	**20.86**/71.45	19.95/67.50	20.28/70.90	20.28/**73.95**
9	21.42/44.75	20.67/43.60	22.81/54.49	22.39/51.45	23.21/62.96	23.10/56.70	23.34/58.17	**24.70/69.87**
10	21.44/46.04	20.68/44.64	22.95/54.40	22.42/52.12	23.19/61.55	23.16/56.73	23.40/57.92	**25.14/66.38**
11	21.15/54.49	20.54/53.06	22.21/59.43	21.94/59.00	22.69/**63.44**	22.45/61.59	22.60/62.23	**23.30**/63.03
12	21.54/49.23	20.72/46.57	22.88/58.47	22.49/55.84	23.16/62.65	23.12/60.04	23.33/61.24	**24.63/69.14**
13	18.67/54.61	19.08/57.77	19.49/56.59	19.48/55.92	21.28/**62.36**	20.52/58.27	20.55/58.35	**21.51**/58.62
14	17.56/37.64	20.40/59.05	19.90/54.84	19.86/55.54	22.44/63.17	21.05/60.78	21.13/60.65	**22.73/66.39**
15	21.92/55.83	21.23/50.26	23.25/65.27	22.97/62.97	23.71/64.35	23.28/65.06	23.43/65.97	**24.16/69.68**
16	22.01/49.57	21.05/44.97	23.65/57.79	23.07/55.64	23.74/59.25	23.56/58.19	23.78/59.15	**25.20/63.16**
17	18.71/46.94	20.93/54.61	21.92/61.04	21.67/60.29	23.25/66.95	22.37/64.54	22.53/65.34	**23.54/69.65**
18	20.94/53.82	20.84/53.68	21.99/58.51	21.82/58.76	23.16/63.37	22.47/62.25	22.61/62.76	**23.40/63.65**
19	21.08/48.72	20.65/48.04	22.20/54.94	21.94/54.05	23.06/62.63	22.51/57.78	22.68/58.56	**23.32/63.10**
20	20.98/58.76	21.90/57.04	21.68/70.47	21.62/68.84	**24.34**/72.14	21.87/71.34	21.95/72.43	22.27/**77.25**
21	21.11/52.54	20.56/51.25	22.25/58.23	21.93/57.20	22.89/63.83	22.50/60.65	22.66/61.42	**23.56/64.70**
22	21.34/50.66	20.80/49.10	22.66/56.97	22.23/55.85	23.26/61.69	22.84/59.19	23.03/60.09	**24.22/63.39**
23	21.63/48.54	21.01/45.93	23.12/58.57	22.60/55.45	23.67/63.72	23.23/59.92	23.46/61.42	**24.96/71.27**
24	19.73/49.56	20.41/53.87	20.81/53.55	20.58/53.77	22.65/**63.52**	21.59/58.78	21.72/59.00	**23.21**/62.80
	$\sigma_n = 75$							
Image	**NLM**	**NLTV**	**WNLM**	**NLMC**	**QNLM**	**QNLTV**	**LPF+QCNLM**	**QBF+QCNLM**
1	19.27/50.76	18.46/48.47	20.10/52.08	19.87/52.90	20.14/52.93	20.44/54.41	20.59/**54.52**	**20.97**/47.34
2	19.86/39.48	19.45/34.75	20.93/47.99	20.52/45.26	21.53/47.23	21.15/50.78	21.36/52.34	**22.25/58.77**
3	19.86/40.02	19.23/36.72	21.09/48.50	20.60/45.49	21.46/50.32	21.37/51.52	21.61/53.33	**22.87/61.97**
4	19.36/39.10	19.10/38.40	20.56/46.34	20.11/43.81	21.17/50.35	21.09/50.45	21.33/51.76	**22.85/60.46**
5	14.25/41.88	18.12/49.41	15.12/46.33	15.54/47.79	19.43/**61.98**	17.11/53.19	17.08/52.59	**19.59**/53.71
6	17.73/34.57	20.99/40.83	18.64/38.47	18.49/38.55	**21.12**/48.69	19.89/46.27	20.06/46.57	21.07/**50.03**
7	17.61/41.18	20.27/48.07	18.44/43.41	18.36/44.61	20.39/**55.40**	19.73/51.09	19.90/51.32	**20.73**/54.94
8	13.68/44.18	18.74/57.90	14.87/48.27	15.35/50.51	18.91/59.72	16.85/55.90	16.81/54.96	**19.03/64.35**
9	19.79/37.69	20.83/48.38	20.99/45.77	20.57/42.93	20.97/49.47	21.42/49.41	21.66/51.01	**22.77/60.50**
10	18.53/31.28	20.90/48.08	19.70/36.39	19.35/35.50	21.03/48.94	20.74/43.70	20.99/44.84	**23.33/58.01**
11	18.40/41.30	20.53/52.68	19.36/45.26	19.09/44.78	20.65/53.01	20.17/50.66	20.35/51.34	**21.59/55.52**
12	19.79/41.89	20.90/50.03	20.90/50.33	20.51/47.41	21.05/50.87	21.24/53.11	21.45/54.53	**22.42/61.59**
13	17.43/46.86	17.14/47.33	18.19/49.06	18.09/48.82	19.57/**54.23**	19.24/48.86	19.34/52.90	**20.27**/53.13
14	16.78/40.81	17.61/45.53	17.50/43.07	17.55/43.84	20.41/**57.16**	19.12/51.03	19.19/50.83	**21.12**/56.37
15	19.65/45.45	19.51/41.87	20.58/53.73	20.27/50.94	21.49/53.33	20.82/55.88	20.98/57.05	**21.64/62.10**
16	19.90/39.24	21.30/35.95	21.20/46.26	20.67/44.01	21.43/48.00	21.49/48.99	21.73/50.22	**23.21/55.80**
17	19.50/48.75	20.96/55.46	20.40/55.47	20.11/53.64	21.08/56.04	20.68/57.90	20.83/58.97	**21.47/61.55**
18	19.55/48.54	20.88/53.20	20.37/53.45	20.14/52.57	21.00/53.55	20.66/55.96	20.80/56.28	**21.22/56.77**
19	19.48/40.65	20.65/49.23	20.50/46.20	20.17/44.97	20.79/50.03	20.87/49.55	21.06/50.55	**21.73/54.31**
20	18.71/49.57	19.00/60.54	19.25/60.80	19.13/57.84	**22.12**/61.25	19.45/62.99	19.53/64.44	19.78/**69.47**
21	19.53/45.01	20.62/52.39	20.57/50.13	20.20/48.76	20.75/52.85	20.90/53.14	21.09/54.14	**21.95/56.73**
22	19.73/42.66	20.91/50.07	20.89/48.63	20.45/46.99	21.04/50.51	21.19/51.50	21.41/52.70	**22.48/55.93**
23	19.85/40.82	21.21/49.79	21.06/49.71	20.58/46.42	21.34/50.55	21.32/52.49	21.55/54.20	**22.71/63.05**
24	19.46/47.47	20.43/52.75	20.44/51.98	20.08/50.86	20.54/53.10	20.73/54.56	20.90/55.37	**21.66/55.68**

$\sigma_n = 75$ Gaussian noise under different denoising algorithms. LPF+QCNLM algorithm and QBF+QCNLM algorithm showed an average increase in FOM of 0.0694 and 0.1029 over the other algorithms and performed well in terms of detail retention, which means that they better preserved the textural detail of the image. Compared to NLM, NLTV, WNLM, NLMC, QNLM, and QNLTV algorithms (Figs.10(c)–(h)), using LPF+QCNLM

Table 8. Denoising results at high noise levels (FOM) using eight algorithms (The best result is in bold)

	$\sigma_n = 55$							
Image	**NLM**	**NLTV**	**WNLM**	**NLMC**	**QNLM**	**QNLTV**	**LPF+QCNLM**	**QBF+QCNLM**
1	0.7223	0.8153	0.7355	0.8125	0.7306	0.7109	0.8160	0.8126
2	0.4760	0.4846	0.5364	0.7579	0.6141	0.7683	0.7645	0.6265
3	0.5271	0.6461	0.6044	0.6607	0.6656	0.7384	0.7247	0.7241
4	0.4522	0.5632	0.5323	0.5425	0.6207	0.6231	0.6851	0.7210
5	0.7979	0.7992	0.8109	0.8308	0.8173	0.8639	0.8617	0.8816
6	0.6213	0.6569	0.6631	0.6476	0.7073	0.6953	0.6876	0.7245
7	0.636	0.7205	0.7052	0.6926	0.7112	0.7508	0.7499	0.7738
8	0.7887	0.8786	0.8864	0.8625	0.9051	0.6346	0.9076	0.8979
9	0.8066	0.7891	0.8361	0.8646	0.8329	0.8952	0.8923	0.8342
10	0.5547	0.6294	0.582	0.6739	0.6478	0.6999	0.7001	0.7022
11	0.6828	0.7203	0.7747	0.7581	0.8336	0.7964	0.8005	0.8505
12	0.8145	0.7388	0.8429	0.8718	0.8431	0.8817	0.8829	0.7949
13	0.6917	0.7442	0.7508	0.7266	0.7534	0.7735	0.8150	0.8178
14	0.5129	0.7437	0.7212	0.7078	0.7521	0.7567	0.7845	0.8203
15	0.7499	0.7664	0.7704	0.8013	0.7864	0.7963	0.8055	0.7756
16	0.4868	0.5294	0.5572	0.5471	0.6126	0.5734	0.5698	0.6259
17	0.6367	0.7082	0.7255	0.6991	0.7387	0.7514	0.7925	0.7983
18	0.6818	0.7201	0.7315	0.7262	0.7562	0.7514	0.7638	0.7915
19	0.6904	0.7645	0.7142	0.7761	0.7471	0.8029	0.8023	0.8005
20	0.5827	0.6306	0.6529	0.6240	0.7152	0.6591	0.6420	0.7165
21	0.5889	0.6662	0.6516	0.6514	0.712	0.6728	0.6646	0.7476
22	0.5676	0.6235	0.6299	0.6263	0.6967	0.6468	0.6392	0.7157
23	0.5628	0.6153	0.5955	0.6897	0.6031	0.6872	0.7086	0.7213
24	0.6566	0.7500	0.7230	0.7043	0.7475	0.7366	0.7812	0.7957
Avg.	0.63703	0.6967	0.6972	0.7189	0.7312	0.7361	0.7600	**0.7696**
	$\sigma_n = 75$							
Image	**NLM**	**NLTV**	**WNLM**	**NLMC**	**QNLM**	**QNLTV**	**LPF+QCNLM**	**QBF+QCNLM**
1	0.6664	0.7330	0.7203	0.7175	0.7604	0.7041	0.7653	0.7816
2	0.4098	0.4875	0.4560	0.4790	0.5183	0.5246	0.5507	0.5570
3	0.4344	0.5510	0.5133	0.5632	0.6467	0.5095	0.6505	0.6844
4	0.4038	0.5162	0.4588	0.4953	0.5773	0.5911	0.5647	0.6272
5	0.6773	0.7250	0.7054	0.7138	0.8003	0.7817	0.8025	0.8407
6	0.5062	0.5945	0.5507	0.5506	0.6099	0.6535	0.6041	0.6559
7	0.5175	0.6973	0.5628	0.5800	0.6451	0.6881	0.6589	0.7150
8	0.7556	0.8176	0.7477	0.7830	0.8300	0.8422	0.8155	0.8664
9	0.6966	0.7901	0.7361	0.7761	0.8221	0.7130	0.8221	0.8003
10	0.3994	0.6313	0.4385	0.4903	0.5809	0.5471	0.5888	0.6385
11	0.5470	0.7070	0.6184	0.6119	0.6949	0.7011	0.7105	0.8140
12	0.7220	0.7612	0.7460	0.8120	0.7771	0.7884	0.8340	0.8430
13	0.5941	0.6648	0.6524	0.6470	0.7049	0.7293	0.7151	0.7730
14	0.5349	0.6210	0.5740	0.5791	0.6630	0.7166	0.6576	0.7431
15	0.6830	0.7151	0.7232	0.7408	0.7641	0.6493	0.7591	0.7736
16	0.4731	0.5701	0.5297	0.4969	0.5565	0.5729	0.6026	0.6361
17	0.5630	0.6690	0.6334	0.6173	0.6461	0.7181	0.6534	0.6790
18	0.6175	0.7280	0.6770	0.6622	0.6916	0.7072	0.6927	0.7213
19	0.4808	0.5917	0.5210	0.5230	0.5558	0.5563	0.5527	0.6081
20	0.4871	0.5969	0.5571	0.5404	0.5500	0.6299	0.5583	0.6148
21	0.4942	0.6453	0.5594	0.5480	0.5805	0.5903	0.6050	0.6582
22	0.5002	0.6104	0.5398	0.5458	0.5747	0.6105	0.5834	0.6208
23	0.4635	0.4878	0.5124	0.6335	0.5801	0.6246	0.6472	0.6339
24	0.5929	0.7278	0.6528	0.6358	0.6701	0.6891	0.7220	0.7399
Avg.	0.5508	0.6516	0.5994	0.6142	0.6583	0.6625	0.6702	**0.70801**

(Fig.10(i)) and QBF+QCNLM algorithms (Fig.10(j)) not only eliminated noise but also better preserved details such as the shape of trees, the boundaries of clouds and the sky, and the reflections of water. In contrast, the QBF+QCNLM denoised image (Fig.10(j)) was the closest to the original image in terms of the contour

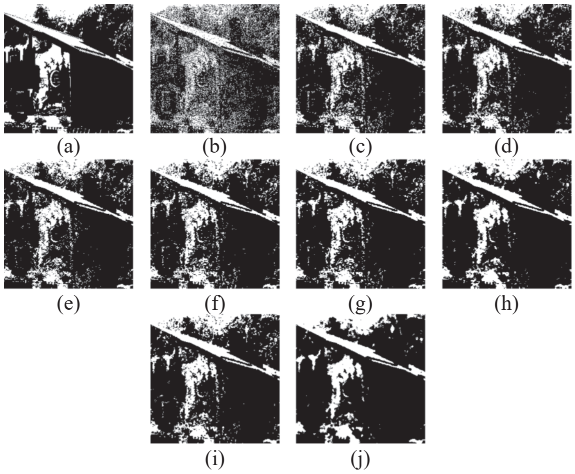

Fig. 9. The visual result image of contours with different denoising algorithms in 24th image from Kodak24. (a) Noise-free 24th image; (b) The 24th image with $\sigma_n = 55$ Gaussian noise; (c)–(j) Denoising by NLM, NLTV, WNLM, NLMC, QNLM, QNLTV, LPF+QCNLM, and QBF+QCNLM, respectively.

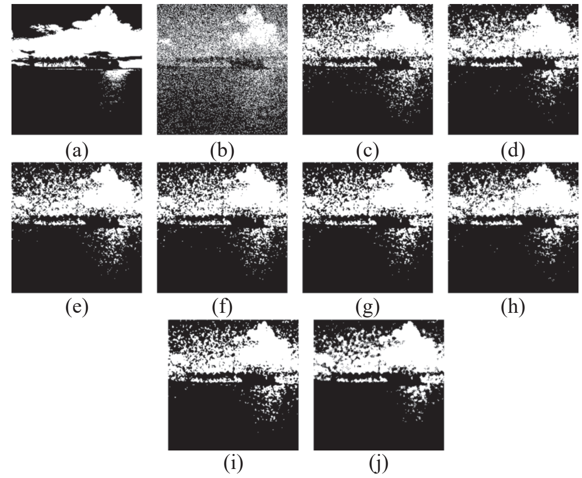

Fig. 10. The visual result image of contours with different denoising algorithms in 16th image from Kodak24. (a) Noise-free 16th image; (b) The 16th image with $\sigma_n = 75$ Gaussian noise; (c)–(j) Denoising by NLM, NLTV, WNLM, NLMC, QNLM, QNLTV, LPF+QCNLM, and QBF+QCNLM, respectively.

lines of the clouds and the trees along the river.

2. Denoising experiments on CBSD68 and McMaster datasets

To further demonstrate the feasibility of our algorithm, we conducted image denoising experiments images with higher resolution CBSD68 (321×481 and 481×321) dataset and McMaster (500×500) dataset. The CBSD68 dataset is the corresponding color version of the greyscale BSD68 dataset and consists of 68 color images. The McMaster dataset is a widely used color decolonization dataset containing 18 cropped images.

We present the final PSNR/SSIM results for the denoising of 110 images from the three image datasets in Table 9.

Table 9. Average PSNR (dB)/SSIM (%) results of seven algorithms for image denoised on CBSD68, McMaster and Kodak24 datasets with noise levels of 10,15, 25, 35, 55 and 75

Dataset	Algorithm	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 25$	$\sigma_n = 35$	$\sigma_n = 55$	$\sigma_n = 75$
CBSD68	NLM	23.88/81.18	23.30/74.37	21.00/59.17	19.39/48.54	19.03/46.34	19.30/47.99
	NLTV	23.92/87.65	22.45/83.07	22.75/66.41	20.20/68.44	21.05/58.65	19.46/48.71
	WNLM	24.82/80.73	24.41/76.47	23.06/64.17	21.54/53.05	19.95/50.09	20.14/52.97
	NLMC	29.42/87.92	26.38/78.62	22.65/63.19	23.82/68.04	20.15/51.97	19.88/51.81
	QNLM	30.16/89.72	27.69/83.26	25.01/73.33	23.97/68.35	**22.52/64.22**	20.43/55.27
	QNLTV	30.47/89.56	27.71/84.13	24.50/72.97	23.93/72.21	21.91/60.73	20.11/53.26
	QCNLM	**31.01/91.44**	**28.99/87.05**	**26.10/77.31**	**25.73/73.98**	22.51/63.90	**20.57/56.08**
McMaster	NLM	27.32/84.40	25.59/74.86	22.10/57.50	19.87/42.71	19.38/41.88	19.34/48.75
	NLTV	27.53/86.53	25.66/82.46	23.61/64.80	21.21/63.30	20.05/56.24	19.24/53.02
	WNLM	28.22/85.72	26.90/79.05	23.71/63.22	22.32/51.73	20.32/46.85	20.13/55.70
	NLMC	30.38/87.97	27.08/77.69	23.16/61.33	24.64/65.23	20.56/49.16	19.89/53.84
	QNLM	31.26/90.19	28.57/83.26	25.75/73.34	24.79/65.47	22.80/63.48	20.38/58.17
	QNLTV	30.73/88.45	26.35/78.23	24.84/72.23	23.03/66.21	22.65/60.76	20.09/57.86
	QCNLM	**32.51/92.86**	**30.28/88.48**	**27.06/78.66**	**26.95/74.27**	**22.83/63.54**	**20.52/59.34**
Kodak24	NLM	25.91/83.79	24.83/74.10	21.99/57.67	19.23/44.70	20.04/49.22	18.65/42.46
	NLTV	27.42/90.29	27.18/80.02	23.31/65.84	23.19/65.05	20.44/51.00	19.90/47.82
	WNLM	29.11/85.05	26.43/77.47	23.36/62.28	21.69/52.10	21.57/56.83	19.65/48.23
	NLMC	29.88/87.39	27.39/79.79	24.25/67.58	25.29/71.27	21.41/56.33	19.40/47.05
	QNLM	30.77/89.65	28.11/82.25	25.42/72.55	25.37/70.93	22.88/64.06	20.80/53.14
	QNLTV	30.42/90.59	27.28/83.05	24.53/70.77	25.12/74.81	22.09/59.36	20.31/52.22
	QCNLM	**31.87/91.87**	**29.88/87.44**	**26.97/78.32**	**26.71/75.74**	**23.57/65.84**	**21.61/57.58**

It is clear that our proposed QCNLM achieved the best PSNR/SSIM than the other competing algorithms, which can be attributed to the fact that the quaternion quasi-Chebyshev distance captured the structural similarity of the color image patch better than the Euclidean distance. At all noise levels, the QCNLM outperformed the NLM, NLTV, WNLM, NLMC, QNLM, and QNLTV algorithms by approximately 4.75 dB, 3.72 dB, 3.24 dB, 2.22 dB, 1.05 dB, and 1.64 dB on average PSNR, and by approximately 16.89%, 8.02%, 12.33%, 8.77%, 3.51%, and 4.23% on average SSIM (Table 9). In terms of preserving image texture information, QCNLM achieved the best FOM at most noise levels compared to other algorithms (Fig.11).

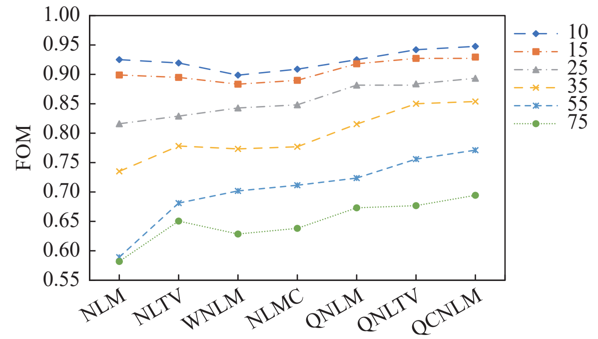

Fig. 11. The results of proposed algorithm are compared with those of other algorithms on FOM.

Compared with NLM, NLTV, WNLM, NLMC, QNLM and QNLTV algorithms, the average FOM gains of the QCNLM algorithm were 0.0907, 0.0562, 0.0603, 0.0528, 0.0255 and 0.0091 (Table 10).

Figs.12–14 give three denoising examples on CBSD68 and McMaster datasets respectively to compare the denoising performance of the QCNLM algorithms with other different algorithms. When images were denoised successively by NLM, NLTV, WNLM, NLMC, QNLM, QNLTV, and QCNLM, it is clear that the noise points gradually decreased, especially in the background, and the various colors gradually became brighter and more vivid. The denoised images by the QCNLM preserved the intrinsic color structure of digital images well and improved the pseudo-texture and edge loss in the smooth areas. Moreover, some specific features of the image were enhanced, e.g., the edges of the castle and its windows were sharper in image “102061” from CBSD68 (Figs.12(c)–(i)), the brown on the horse’s back was brighter in image “220075” from CBSD68 (Figs.13(c)–(i)) and the various colors on the white cloth were more intuitive in image “5” from McMaster (Figs.14(c)–(i)).

Table 10. Average FOM results of seven algorithms for image denoised on CBSD68, McMaster and Kodak24 datasets with noise levels of 10,15, 25, 35, 55 and 75

Algorithm	$\sigma_n = 10$	$\sigma_n = 15$	$v_n = 25$	$\sigma_n = 35$	$\sigma_n = 55$	$\sigma_n = 75$	Avg.
NLM	0.9250	0.8989	0.8160	0.7350	0.5891	0.5818	0.7576
NLTV	0.9194	0.8948	0.8290	0.7782	0.6811	0.6505	0.7921
WNLM	0.8987	0.8833	0.8428	0.7732	0.7018	0.6285	0.7880
NLMC	0.9088	0.8899	0.8482	0.7769	0.7114	0.6381	0.7955
QNLM	0.9249	0.9180	0.8816	0.8154	0.7236	0.6730	0.8228
QNLTV	0.9419	0.9272	0.8838	0.8500	0.7559	0.6767	0.8392
QCNLM	**0.9476**	**0.9294**	**0.8933**	**0.8538**	**0.7711**	**0.6943**	**0.8483**

3. Complexity analysis

We compared the actual theoretical complexity between the QCNLM algorithm and six known algorithms, including NLM, NLTV, WNLM, NLMC, QNLM and QNLTV. The main computationally intensive part of these algorithms lies in the image patch similarity metric by using Euclidean distance or quasi-Chebyshev.

Since the Euclidean distance is to calculate a weighted average of three channels of **all** pixels, while the quasi-Chebyshev distance is to calculate a weighted average of three channels for **each** pixel and then find the maximum value, it means that to calculate the Euclidean distance needs more operations than quasi-Chebyshev distance, leading to that our QCNLM algorithm has lower algorithm complexity than six known algorithms. The NLTV/WNLM algorithms are hybrid of NLM and TV/wavelet, leading to an increase in algorithm complexity. Since the NLM/NLTV/WNLM algorithms process the RGB channels separately while NLMC/QNLM/QNLTV/QCNLM algorithms process the RGB channels as a whole, the NLM/NLTV/WNLM algorithms have higher algorithm complexity than the other four algorithms. When dealing with high level noises, pre-processing with LPF or QBF at high noise level increases algorithm complexity. In addition, the large size of image patches and search areas also increases algorithm complexity.

We tested the running time of various denoising algorithms. We ran denoising experiments using MATLAB R2020a, on a laptop 2.40 GHz Intel Core i5-1135G7 CPU with 8 GB@3200 MHz DDR4 memory.

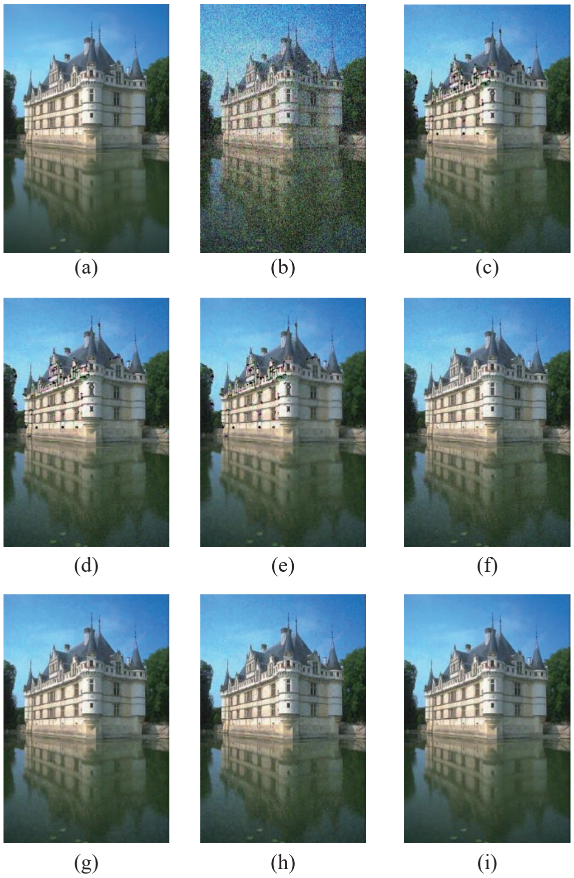

Fig. 12. The visual result image of two denoising algorithms in image “102061” from CBSD68. (a) Noise-free image “102061”; (b) The image “102061” destroyed by Gaussian noise with low noise level ($\sigma_n = 15$); (c)–(i) Denoising by NLM, NLTV, WNLM, NLMC, QNLM, QNLTV, and QCNLM, respectively.

The codes of NLM[*1], NLTV[*2], NLMC[*3] and the wavelet threshold[*4] algorithms can be accessed from windows.

By using this and original NLM codes, it is easy to implement WNLM. The Q-lib library[*5] provides the code for the operation of quaternion vectors and matrices.

By using this and original NLM/NLTV codes, it is easy to implement QNLM and QNLTV and our QCNLM algorithm.

The average running time of each denoising algorithm is shown in Table 11 and Fig.15. Our proposed QCNLM demonstrated the fastest denoising with an average execution time of 13.87 seconds, and with an average improvement of 34.83% over other algorithms.

Fig. 13. The visual result image of two denoising algorithms in image “220075” from CBSD68. (a) Noise-free image “220075”; (b) The image “220075” destroyed by Gaussian noise with middle noise level ($\sigma_n = 35$); (c)–(i) Denoising by NLM, NLTV, WNLM, NLMC, QNLM, QNLTV, and QCNLM, respectively.

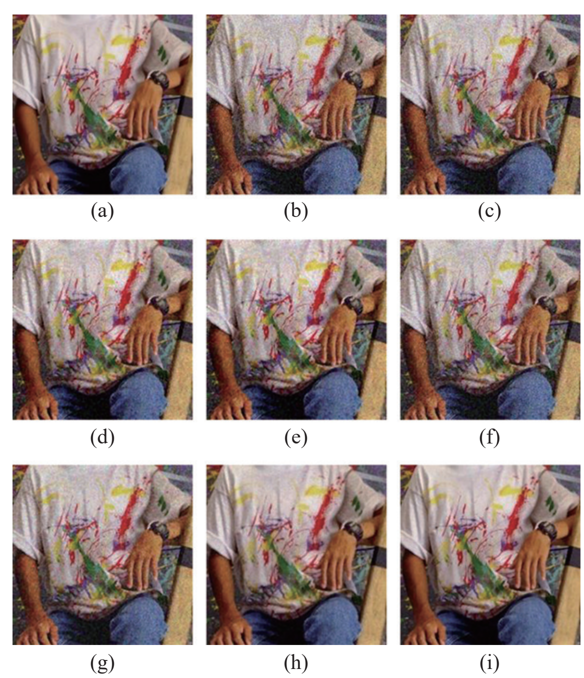

Fig. 14. The visual result image of two denoising algorithms in image “5” from McMaster. (a) Noise-free image “5”; (b) The image “5” destroyed by Gaussian noise with high noise level ($\sigma_n = 55$); (c)–(i) Denoising by NLM, NLTV, WNLM, NLMC, QNLM, QNLTV, and QCNLM, respectively.

4. Parametric effects

To further demonstrate parametric effects of the

[*1] https://github.com/sepidsh/Image_denoising_NLM
[*2] https://github.com/Tinrry/BOS_NLTV_v1
[*3] https://github.com/xavirema/nlmc
[*4] https://github.com/helderc/WaveletTransformShrinkThreshold
[*5] https://sourceforge.net/projects/qtfm/

Table 11. Average running time (in seconds) of each noise reduction algorithm with different noise levels

Algorithm	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 25$	$\sigma_n = 35$	$\sigma_n = 55$	$\sigma_n = 75$	Avg.
NLM	10.53	12.65	12.67	34.38	33.80	33.95	22.99
NLTV	10.75	14.14	18.30	37.80	35.52	35.52	25.33
WNLM	10.63	11.70	15.97	38.09	29.22	29.47	22.51
NLMC	7.30	7.95	10.02	26.63	21.70	21.80	15.90
QNLM	8.07	10.05	10.44	33.27	26.65	29.05	19.58
QNLTV	10.96	11.06	13.81	37.98	28.28	26.28	21.39
QCNLM	**6.58**	**6.68**	**9.20**	**20.78**	**19.89**	**20.09**	**13.87**

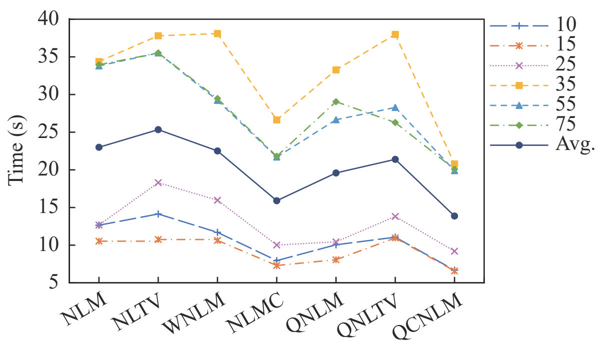

Fig. 15. Denoising results (running time(s)) using seven algorithms.

proposed QCNLM model, we performed two sets of color image denoising experiments with different image patch size w and denoising parameter h, where the test images were 24 color images from the Kodak24 image dataset (Fig.2).

In the first set of experiments, image patches of sizes $w = 3, 5$, and 7 were used to test the performance of various denoising algorithm under different noise levels (see Table 12). It demonstrated that our QCNLM consistently outperforms NLM, NLTV, WNLM, NLMC, QNLM, and QNLTV at all noise levels. In details, a) For an image patch size $w = 3$, the average PSNR/SSIM gain of QCNLM over the other six algorithms is 4.52 dB/19.60%, 4.92 dB/14.42%, 4.01 dB/19.02%, 1.99 dB/6.61%, 1.31 dB/5.01%, and 1.43 dB/6.07%; b) For an image patch size $w = 5$, the average PSNR/SSIM gain of QCNLM over the other six algorithms is 6.77 dB/27.09%, 6.30 dB/23.73%, 4.16 dB/21.93%, 2.13 dB/9.51%, 1.97 dB/7.28%, and 2.44 dB/7.54%. c) For an image patch size $w = 7$, the average PSNR/SSIM gain of QCNLM over the other six algorithms is 5.79 dB/20.27%, 4.98 dB/14.71%, 3.47 dB/14.92%, 4.24 dB/14.80%, 2.81 dB/5.58%, and 1.87 dB/1.95%.

In the second set of denoising experiments, different denoising parameters h were tested (Fig.16). With the increase of the denoising parameter h, better PANR and SSIM denoising results were achieved. Our proposed QCNLM algorithm also outperformed six known algorithms in terms of PSNR and SSIM.

V. Conclusions and Discussions

The non-local means (NLM) denoising algorithm for gray images is asymptotically optimal under a generic statistical image algorithm. By introduction of quaternion representation of color images, the original NLM algorithm for gray image denoising is extended into the quaternion non-local means (QNLM) algorithm for color image denoising. In this paper, we propose the quaternion quasi-Chebyshev non-local means (QCNLM) for color image denoising by incorporating quaternion quasi-Chebyshev distance into quaternion non-local means (QNLM). Our proposed quaternion quasi-Chebyshev distance demonstrated strong capacity in capturing the similarity between image patches and reduced the amount of calculation significantly when compared with Euclidean distance, leading to the finding that denoising performance of QCNLM was better than the known algorithms (NLM, NLTV, MNLM, NLMC, QNLM, QNLTV).

Table 12. Average denoising results of seven algorithms with different patch size and noise level

Patch size	Noise level	NLM	NLTV	WNLM	NLMC	QNLM	QNLTV	QCNLM
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
3	20	27.36/63.66	26.20/68.66	26.81/64.59	30.08/80.32	30.13/78.89	29.45/82.61	**31.20/83.26**
	30	24.17/67.14	23.86/71.70	25.15/67.60	27.31/82.53	27.89/83.11	28.20/76.43	**29.42/86.15**
	50	23.74/65.82	24.02/71.80	24.85/66.19	25.47/72.75	26.91/78.38	26.90/78.17	**28.23/86.03**
5	20	20.26/54.39	20.86/51.47	22.94/56.65	26.73/74.63	25.61/74.04	25.28/75.22	**28.20/80.42**
	30	20.19/54.28	20.65/50.90	22.85/56.36	26.15/75.96	25.35/72.21	25.30/75.38	**27.43/78.70**
	50	20.07/54.16	20.41/50.38	22.54/55.21	21.56/54.92	23.96/65.93	22.93/60.81	**25.21/74.91**
7	20	20.15/49.77	20.74/57.68	22.45/54.77	21.07/52.65	21.89/56.26	25.14/74.68	**27.06/77.92**
	30	16.50/32.36	17.39/36.63	18.87/38.15	19.46/43.59	21.97/54.70	19.83/47.63	**23.16/59.37**
	50	16.27/32.13	17.21/36.61	18.54/37.39	17.02/34.41	18.01/**47.35**	19.69/46.91	**20.07**/37.79

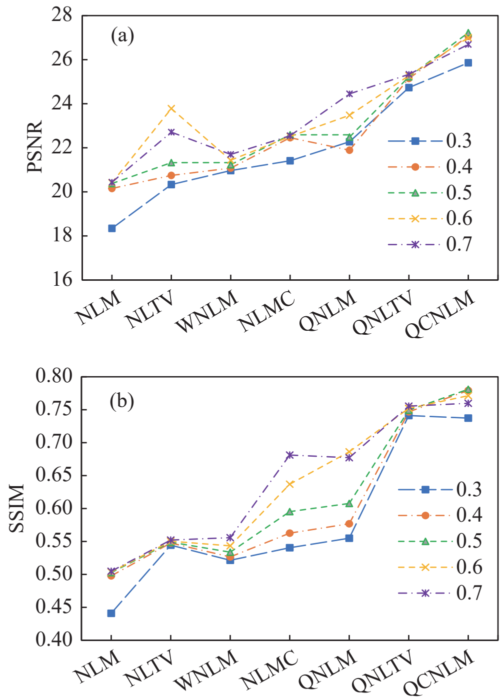

Fig. 16. Denoising PSNR (dB)/SSIM (%) results using seven algorithms with different denoising parameter.

Denoising experimental simulations of 110 color images of landscapes, people and buildings demonstrated that the proposed QCNLM not only achieved the best PSNR, SSIM and FOM, but also better preserved complex textures and curves and had the best visual experience. For low noise levels ($\sigma_n = 10, 15, 25, 35$), compared with NLM, NLTV, MNLM, NLMC, QNLM and QNLTV algorithms, the average PSNR/SSIM gains of the QCNLM algorithm were 3.37 dB/5.01%, 3.42 dB/8.09%, 3.09 dB/12.0%, and 3.76 dB/12.8%, and the average FOM index gains were approximately 0.0227, 0.0114, 0.0117 and 0.0384. For high level noise ($\sigma_n = 55, 75$), we introduced the Quaternion bilateral filtering (QBF) and used it to preprocess the noisy image. The combination of QBF and QCNLM algorithms could retain well the inherent color structure, reduced the phenomenon of false texture and edge loss in the smooth area of the denoised image and enabled a better visual perception denoising effect. The average PSNR, SSIM and FOM values of QBF+QCNLM were approximately 1.15 dB, 4.75% and 0.0236 higher than the LPF+QCNLM algorithm, and 2.48 dB, 9.78% and 0.0523 higher than the other six algorithms.

Although our QCNLM outperformed NLM, NLTV, MNLM, NLMC, QNLM and QNLTV, denoised images by our QCNLM still demonstrated artifacts and slight discontinuities in some boundaries of small patches. In the future, we will consider coupling quaternion quasi-Chebyshev distances into neural-network-based denoising models [31] to better capture the similarity of image patches [32] and then achieve better denoising performance. At the same time, we also plan to extend our algorithm to color image segmentation [33], protrusion detection and deblurring [34].

References

[1] J. B. Martens and L. Meesters, "Image dissimilarity," *Signal Processing*, vol.70, no.3, pp.155–176, 1998.

[2] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol.27, no.1, pp.13–18, 1979.

[3] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol.32, no.6, pp.1109–1121, 1984.

[4] J. S. Goldstein, I. S. Reed, and L. L. Scharf, "A multistage representation of the Wiener filter based on orthogonal projections," *IEEE Transactions on Information Theory*, vol.44, no.7, pp.2943–2959, 1998.

[5] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Transactions on Automatic Control*, vol.45, no.5, pp.910–927, 2000.

[6] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the 6th International Conference on Computer Vision*, Bombay, India, pp.839–846, 2002.

[7] A. Buades, B. Coil, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol.4, no.2, pp.490–530, 2005.

[8] G. Gilboa and S. Osher, "Nonlocal evolutions for image regularization," in *Proceedings of SPIE 6498, Computational Imaging V*, San Jose, United States, pp.64980U, 2007.

[9] B. Goossens, H. Luong, J. Aelterman, *et al.*, "A GPU-accelerated real-time NLMeans algorithm for denoising color video sequences," in *Proceedings of the 12th International Conference on Advanced Concepts for Intelligent Vision Systems*, Sydney, Australia, pp.46–57, 2010.

[10] A. Khmag, S. A. R. Al Haddad, R. A. Ramlee, *et al.*, "Natural image noise removal using nonlocal means and hidden Markov models in transform domain," *The Visual Computer*, vol.34, no.12, pp.1661–1675, 2018.

[11] M. Diwakar and M. Kumar, "CT image denoising using NLM and correlation-based wavelet packet thresholding," *IET Image Processing*, vol.12, no.5, pp.708–715, 2018.

[12] K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*. Springer, Berlin, Heidelberg, 2000.

[13] H. Y. Yang, Y. Zhang, P. Wang, *et al.*, "A geometric correction based robust color image watermarking scheme using quaternion exponent moments," *Optik*, vol.125, no.16, pp.4456–4469, 2014.

[14] X. Y. Wang, Z. F. Wu, L. Chen, *et al.*, "Pixel classification based color image segmentation using quaternion exponent moments," *Neural Networks*, vol.74, pp.1–13, 2016.

[15] C. J. Chen, Y. Xu, and X. K. Yang, "User tailored colorization using automatic scribbles and hierarchical features," *Digital Signal Processing*, vol.87, pp.155–165, 2019.

[16] Ö. N. Subakan and B. C. Vemuri, "A quaternion framework for color image smoothing and segmentation," *International Journal of Computer Vision*, vol.91, no.3, pp.233–250, 2011.

[17] B. J. Chen, H. Z. Shu, H. Zhang, *et al.*, "Quaternion Zernike moments and their invariants for color image analysis and object recognition," *Signal Processing*, vol.92, no.2, pp.308–318, 2012.

[18] A. Buades, B. Coil, and J. M. Morel, "Non-local means denoising," *Image Processing on Line*, vol.1, pp.208–212, 2011.

[19] A. Buades, B. Coll, and J. M. Morel, "Image processing on line: nonlocal means denoising," Available at: *http://www.ipol.im/pub/algo/bcm non local means denoising/*, 2022.

[20] Q. Li, J. F. Teng, Q. M. Zhao, *et al.*, "Wavelet domain wiener filter and its application in signal denoising," in *Proceedings of the 3rd International Conference on Wavelet Analysis and Its Applications (WAA)*, Chongqing, China, pp.839–846, 2003.

[21] F. Z. Zhang, "Quaternions and matrices of quaternions," *Linear Algebra and Its Applications*, vol.251, pp.21–57, 1997.

[22] B. J. Chen, Q. S. Liu, X. M. Sun, *et al.*, "Removing Gaussian noise for colour images by quaternion representation and optimisation of weights in non-local means filter," *IET Image Processing*, vol.8, no.10, pp.591–600, 2014.

[23] G. H. Wang, L. Yang, X. Wei, *et al.*, "An improved nonlocal means filter for color image denoising," *Optik*, vol.173, pp.157–173, 2018.

[24] Z. G. Jia, Q. Y. Jin, M. K. Ng, *et al.*, "Non-local robust quaternion matrix completion for large-scale color image and video inpainting," *IEEE Transactions on Image Processing*, vol.31, pp.3868–3883, 2022.

[25] X. Y. Li, Y. C. Zhou, and J. Zhang, "Quaternion non-local total variation for color image denoising," in *Proceedings of 2019 IEEE International Conference on Systems, Man and Cybernetics*, Bari, Italy, pp.1602–1607, 2019.

[26] R. Franzen, "Kodak lossless true color image suite," Available at: *http://r0k.us/graphics/kodak*, 1999.

[27] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, pp.860–867, 2005.

[28] L. Zhang, X. L. Wu, A. Buades, *et al.*, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *Journal of Electronic Imaging*, vol.20, no.2, article no.023016, 2011.

[29] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol.9, no.3, pp.81–84, 2002.

[30] B. Magnier, "Edge detection evaluation: A new normalized figure of merit," in *Proceedings of 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, pp.2407–2411, 2019.

[31] A. Khmag, "Additive Gaussian noise removal based on generative adversarial network model and semi-soft thresholding approach," *Multimedia Tools and Applications*, vol.82, no.5, pp.7757–7777, 2023.

[32] Y. Peng, "Quantitative method based on cotangent similarity degree in three-valued Łukasiewicz logic," *Chinese Journal of Electronics*, vol.30, no.1, pp.134–144, 2021.

[33] Y. F. Li, Q. J. Zhao, W. B. Zhang, *et al.*, "A simultaneous cartoon-texture image segmentation and image decomposition method," *Chinese Journal of Electronics*, vol.29, no.5, pp.906–915, 2020.

[34] M. Li and C. Xu, "Variational image restoration and decomposition in shearlet smoothness spaces," *Chinese Journal of Electronics*, vol.26, no.5, pp.1017–1021, 2017.

XU Xudong was born in 1995. She received the B.S. degree from Beijing University of Technology and Industry, China, and M.S. degree from University of Science and Technology Beijing, China. She is currently a Ph.D. student in the School of Mathematics at Shandong University, China. Her research interests include image processing and remote sensing image classification. (Email: 202120281@sdu.edu.cn)

ZHANG Zhihua (corresponding author) is a Taishan Distinguished Professor of Big Data in School of Mathematics, Shandong University, Jinan, China. He has published 6 first-authored monographs and over 50 first-authored articles. Prof. Zhang is an Associate Editor of *EURASIP Journal on Advances in Signal Processing*.
(Email: zhangzhihua@sdu.edu.cn)

M. James C. Crabbe is a Professor and Fellow at Oxford University, Oxford, UK and a Visiting Professor at the University of Reading, UK. He was on the Executive Committee of the UK Deans of Science and was elected a Distinguished Fellow of the Institute of Data Science and Artificial Intelligence.

(Email: james.crabbe@wolfson.ox.ac.uk)