



Image denoising using multidirectional gradient domain

Xiaobo Zhang¹

Received: 3 October 2020 / Revised: 11 April 2021 / Accepted: 24 June 2021 /

Published online: 12 July 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

This paper presents a new two-step image denoising method termed multidirectional gradient domain image denoising (MGDID). In each step, unlike previous gradient domain designs, the multidirectional gradient domain information is used to represent the noise component so that the more directional image features are extracted. The Gaussian pre-filter is carried out in the square gradient coefficients. The nonlinear remedied factor is adopted to modify the denoising amount. The whole denoising process originates from classical nonlocal means (NLM) and nonlinear diffusion. MGDID takes full advantage of ability of NLM to better process the image with the rich repetitive features and the denoising scheme of relatively simplicity and efficiency of nonlinear diffusion. Experimental results show MGDID is superior to the related gradient domain methods and NLM methods in peak signal-to-noise ratio (PSNR), mean structural similarity (MSSIM) and visual performance. For example, for Barbara image with the rich repetitive texture feature, MGDID outperforms classical NLM from 0.33 dB to 1.66 dB in PSNR. Usually, classical NLM wins the local adaptive layered Wiener filter (a state-of-the-art gradient domain method) more than 0.44 dB for Barbara. In addition, MGDID is also very efficient compared to the related methods.

Keywords Image denoising · Nonlocal means (NLM) · Nonlinear diffusion · Gradient domain

1 Introduction

There are many tasks in the field of image processing. A literature often focuses on a specific task. For example, the literature [7] designed the multiple images steganography algorithm by studying adaptive payload distribution. The literature [8] proposed a convolutional neural network based forensic framework for detecting the image operations. Denoising is then one

✉ Xiaobo Zhang
zhangxiaobo419@126.com

¹ Institute of Graphics and Image Processing, Xianyang Normal University, Xianyang 712000, China

of the most fundamental problems in the field of image processing. Currently, a variety of denoising methods have been proposed in the different forms. Among these methods, window-based methods have achieved success. The classical window-based methods have the local Wiener filter method [11] and the nonlocal means (NLM) filter method [1]. They are the representative methods of the window-based local ones and the window-based nonlocal ones, respectively. The common point of two methods is that the denoised result of the processed pixel is obtained by employing the information of the pixels in a neighborhood window centered at the processed pixel. The different point of two methods is that the weights of pixels in the window are the same commonly in the window-based local methods while the weights of pixels in the window are different customarily in the window-based nonlocal methods. Currently, two methods have attracted significant attention. The window-based local methods are very efficient. There have been many variants [2, 4, 13, 18–21]. The methods in [4, 18–21] are the development of local Wiener filter. The methods in [4, 18] operate in the wavelet domain. The methods in [19–21] can also be seen as variants of nonlinear diffusion in [13]. In fact, nonlinear diffusion method (PMAD) is popular and has been attracting attention since it was proposed by Perona and Malik in [13]. For example, Gaussian filter acts on the image domain to overcome the theoretical and practical limitations of the PMAD model [2]. To solve PMAD model, a new algorithm based on mixed finite element method was proposed in [5]. Recently, a hybrid diffusion framework was established so that the desirable mathematical properties are obtained for PMAD model [9]. On the other hand, the window-based nonlocal methods [3, 6, 10, 12, 14, 15, 17] obtain better performance. But it has been found that the window-based nonlocal methods have usually high computational load. Therefore, some improvements [6, 15] are done to accelerate the window-based nonlocal methods. In addition, the parameters on window-based nonlocal methods are also researched. For example, the weight of the center pixel is modified so that the better results are obtained [12, 14, 17]. In [14], the center weight based on the Stein's unbiased risk estimate principle was proposed. It is a global center weight which uses the constant weight for all pixels. In [17], a new local James-Stein type center pixel weight (LJSCPW) was developed. LJSCPW is locally tuned for each image pixel. In [12], the novel local center pixel weight estimation methods using Baranchik's minimax estimator were proposed based on the LJSCPW method. In [3], NLM is improved by the shape-adaptive patches.

In view of good performance of the window-based nonlocal methods and efficiency of the window-based local methods, this paper presents a new image denoising method based on the neighborhood window centered at the processed pixel. The proposed method is different from the previous gradient domain methods. It is called multidirectional gradient domain image denoising (MGDID). This is because that it operates in the multidirectional gradient domain so that the image features can be better captured. And the time step size is adaptive for image contents. The proposed method is different from the NLM filter. This is because that the filter result is obtained in term of the iterative implementation of nonlinear diffusion. Experiment results show that the proposed MGDID obtains good results but is very efficient because of possessing the advantages of the window-based nonlocal and local methods.

The paper is organized as follows. Section 2 first reviews window-based gradient domain layered denoising method and window-based nonlocal method. And then, the derivation of the proposed method is presented. At last, the adjustment strategy of the parameters for the proposed method is described in detail. The experimental results are presented in Section 3. The simulation results show the proposed method achieves the desired effects. The paper is concluded in Section 4.

2 The proposed method

Assume an original image is degraded by additive noise and the noise is signal independent. The typical image degradation model at (i, j) in a two-dimensional coordinate can be written as

$$I_0(i, j) = I(i, j) + n(i, j) \quad (1)$$

where I , I_0 , and n represent the original image, the observed image, and the additive Gaussian random noise with zero mean and variance σ^2 , respectively. The aim here is to restore the original pixel by using neighbor pixels centered at the processed pixel. In the following, the several respects on the proposed method are detailed.

2.1 Gradient domain layered denoising

Gradient domain image denoising based on nonlinear diffusion was proposed in [2, 13, 19–21]. Among these methods, the gradient domain local adaptive layered Wiener filter (LALWF) in [20] achieves the very good performance. This method is formed by researching the statistical property of gradient coefficients. In this method, the local Wiener filter [11] was successfully embedded into gradient domain. In term of [20], the scheme of gradient domain LALWF can be written as:

$$I_k(i, j) = I_{k-1}(i, j) + \beta \sum_{u=-l}^l \sum_{v=-l}^l \nabla \hat{n}_{k-1}^{u,v}(i, j) \quad (2)$$

$$\nabla \hat{n}_{k-1}^{u,v}(i, j) = \text{threshold} \cdot \nabla I_{k-1}^{u,v}(i, j) \quad (3)$$

where $\nabla I_{k-1}^{u,v}(i, j) = I_{k-1}(i+u, j+v) - I_{k-1}(i, j)$ ($u = -l, -l+1, \dots, -1, 1, \dots, l$ and $v = -l, -l+1, \dots, -1, 1, \dots, l$) represent the differences of different distances in the different directions centered at (i, j) in the “square” search window, respectively. $l \geq 0$ is an integer. β is a constant. I_k represents the k th denoised image. The $\nabla I_k^{u,v}(i, j)$ is also called the gradient coefficient for image I_k . The computations of gradient coefficient of other images are the same as that of I_k . The (2) and (3) can also be used to describe the traditional gradient domain scheme in form. The difference lies in that *threshold* is taken as the local Wiener filter in gradient domain LALWF scheme. Obviously, the scheme (2) adopts the iterative scheme to remove noise layer by layer. The preceding step denoising result is the input of the current step denoising. The initial noisy image is input of the first step denoising. In the traditional gradient domain denoising including gradient domain LALWF, the pixels at (i, j) , $(i+1, j)$, $(i-1, j)$, $(i, j+1)$ and $(i, j-1)$ are included in the (search) window. That is to say, the “+” shape (search) window is used in traditional gradient domain denoising. The advantage of the layered denoising lies in that the denoising amount can be flexibly grasped so that the features of image are preserved at the controllable degree.

The proposed method will adopt the layered denoising scheme. But, unlike the previous gradient domain, the multidirectional (search) window is used. For the “square” multidirectional (search) window, please see Fig. 1 ($l = 2$). Multidirectional (search) window can be used to extract the more directional image features. In Fig. 2, clean House image is used for demonstrating the ability of the proposed method to extract multidirectional information. Image feature maps extracted by using multidirectional (search) window with $l = 1$ are shown.

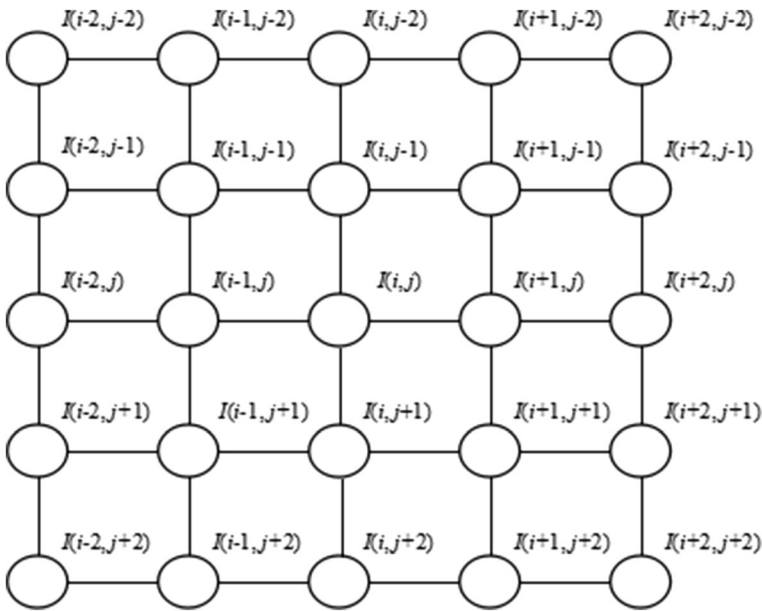


Fig. 1 The “square” search window

That is to say, the “square” 8 direction (search) window is used in Fig. 2. Obviously, compared to the “+” shape search window with 4 directions, multidirectional (search) window with the “square” 8 direction (search) window extracts more image information (Figs. 2b, d, f and h are the feature maps with “+” shape search window in four different directions).

In Fig. 2, for the sake of comparison, the feature map $f_{\nabla I^{1,0}}$ in the (1, 0) direction is given by

$$f_{\nabla I^{1,0}}(i, j) = \frac{(\nabla I^{1,0}(i, j))^2 - \text{Min}(\nabla I^{1,0})^2}{\text{Max}(\nabla I^{1,0})^2 - \text{Min}(\nabla I^{1,0})^2} \cdot 255 \tag{4}$$

where $\text{Min}(\nabla I^{1,0})^2$ and $\text{Max}(\nabla I^{1,0})^2$ are the minimum and maximum “signal” gradient (difference) maps of the clean image House in the (1, 0) direction. The computations of feature maps of different directions of other image are the same as above.

In the design of denoising algorithm, the pre-filter has been successfully used in wavelet transform domain and nonlinear diffusion such as gradient domain for image denoising. So, to better implement the denoising process, the pre-filter way is also adopted in the proposed method. The Gaussian filter is used in the stage of the pre-filter of the proposed method. It operates in the square gradient coefficient. Gaussian filter is used because it assigns the processed pixel the higher weight so that the edges can be retained in the process of denoising.

Gradient domain LALWF employs local Wiener filter to obtain good performance. Local Wiener filter shrinkage function is as follows:

$$\text{threshold} = \frac{\sigma_{k-1}^2}{\sigma_{k-1}^2 + \max(q_{k-1}(i, j) - \sigma_{k-1}^2, 0)} \tag{5}$$

where the $\max(q_{k-1}(i, j) - \sigma_{k-1}^2, 0)$ is “signal” variance at iteration k , the $q_{k-1}(i, j)$ in size $(2l_1 + 1) \times (2l_1 + 1)$ local (similar) window is computed as

Fig. 2 Feature maps for House (256 × 256) image. Zoom into file for a better view. **(a)** noise-free House image. **(b)** (1,0)direction. **(c)** (1,-1)direction. **(d)** (0,-1) direction. **(e)** (-1,-1)direction. **(f)** (-1,0)direction. **(g)** (-1,1)direction. **(h)** (0,1)direction. **(i)** (1,1)direction

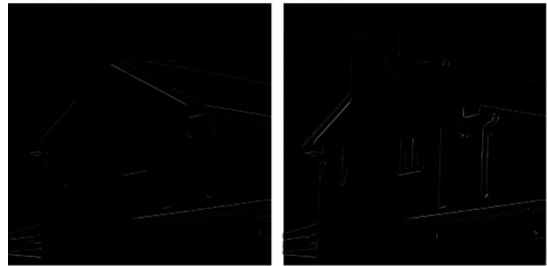


(a)



(b)

(c)



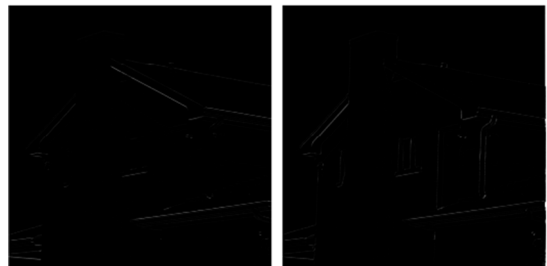
(d)

(e)



(f)

(g)



(h)

(i)

$$q_{k-1}(i, j) = \frac{1}{(2l_1 + 1)^2} \sum_{i1=-l_1}^{l_1} \sum_{j1=-l_1}^{l_1} (\nabla I_{k-1}^{u,v}(i + i1, j + j1))^2 \tag{6}$$

From above, the conclusion can be drawn that the computation of local “signal” variance is based on the square gradient coefficient. This is the vital step so that Wiener filter can get the good performance. And then, the local Wiener filter shrinkage function is formed. Therefore, the proposed pre-filter way takes the advantage. That is that the proposed method approximately uses the advantages of local Wiener filter because the square gradient coefficient will be integrated into the following shrinkage function. This is different from the previous gradient domain filter. In the previous nonlinear diffusion gradient domain implementation, for example, the method in [2], the image domain Gaussian filter is used, and which is equivalent to the gradient domain Gaussian filter on the gradient coefficient.

2.2 Nonlocal means (NLM)

In the NLM algorithm [1], the restored intensity $\widehat{I}(i, j)$ is the weighted average of all the pixel intensity values of the processed pixel neighborhood in the noisy image. This method is expressed as:

$$\widehat{I}(i, j) = \sum_{(i', j') \in N(i, j)} \omega\left(\left(\begin{matrix} i' \\ j' \end{matrix}\right), (i, j)\right) I_0(i, j) \tag{7}$$

where $N(i, j)$ represents the neighborhood at (i, j) and is called the search window. The weight $\omega\left(\left(\begin{matrix} i' \\ j' \end{matrix}\right), (i, j)\right)$ depending on the similarity between $I_0(i, j)$ and $I_0(i', j')$ satisfies $0 \leq \omega\left(\left(\begin{matrix} i' \\ j' \end{matrix}\right), (i, j)\right) \leq 1$ and $\sum_{(i', j') \in N(i, j)} \omega\left(\left(\begin{matrix} i' \\ j' \end{matrix}\right), (i, j)\right) = 1$. The similarity between $I_0(i, j)$ and $I_0(i', j')$ depends

on the similarity of the intensity gray level vectors $I_0(N_s(i, j))$ and $I_0(N_s(i', j'))$. Here, $N_s(i, j)$ and $N_s(i', j')$ represents the neighborhoods at (i, j) and (i', j') , respectively. This similarity is measured as a decreasing function of the weighted Euclidean distance $\|I_0(N_s(i, j)) - I_0(N_s(i', j'))\|_{G_\rho}^2 = G_\rho * \|I_0(N_s(i, j)) - I_0(N_s(i', j'))\|^2$ where G_ρ is the Gaussian kernel with standard deviation ρ . The weights are computed as:

$$\omega\left(\left(\begin{matrix} i' \\ j' \end{matrix}\right), (i, j)\right) = \frac{1}{C(i, j)} e^{-\frac{\|I_0(N_s(i, j)) - I_0(N_s(i', j'))\|_{G_\rho}^2}{h^2}} \tag{8}$$

where $C(i, j)$ is the normalizing constant

$$C(i, j) = \sum_{(i', j') \in N(i, j)} e^{-\frac{\|I_0(N_s(i, j)) - I_0(N_s(i', j'))\|_{G_\rho}^2}{h^2}} \tag{9}$$

and the parameter h acts as a degree of filtering. It controls the decay of the exponential function.

Here, let $O_h = e^{-\frac{sh^2}{h^2}}$ ($sh = \sqrt{\|I_0(N_s(i, j)) - I_0(N_s(i', j'))\|_{G_\rho}^2}$), and O_h is called the shrinkage coefficient. sh describes the difference degree between two pixels $I_0(i, j)$ and $I_0(i', j')$. And then

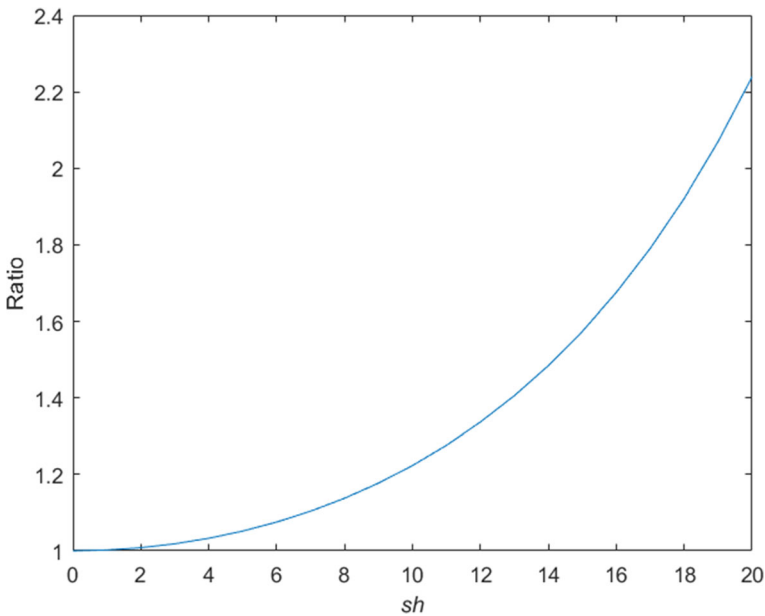


Fig. 3 Ratio = O_{18}/O_{14} varies with the difference degree sh between two pixels ($O_h = e^{-\frac{sh^2}{h^2}}$, $sh = \sqrt{\|I_0(N_s(i,j)) - I_0(N_s(i',j'))\|_{G_p}^2}$)

let Ratio = O_{18}/O_{14} . Figure 3 presents the picture of Ratio changing with sh . A conclusion can be drawn. The change of h will lead to greater change of the shrinkage coefficient O_h with larger sh . Larger sh often describes more important features of image. Larger h can remove more noise, but may also destroy image features. So, the proper h is very important for denoising performance. In the classical NLM [1], h is σ so that the good result is obtained.

2.3 Derivation for the proposed multidirectional gradient domain image denoising (MGDID)

In fact, after a series of manipulations, the nonlocal scheme (7) can be written as

$$\hat{I}(i, j) = I_0(i, j) + \frac{1}{C(i, j)} \sum_{(i', j') \in N_0(i, j)} e^{-\frac{\|I_0(N_s(i, j)) - I_0(N_s(i', j'))\|_{G_p}^2}{h^2}} (I_0(i', j') - I_0(i, j)) \quad (10)$$

where $N_0(i, j)$ represents the hollow neighborhood at (i, j) . That is, (i, j) is not included in $N_0(i, j)$. From (10), it can be seen that (7) can be considered as one step gradient domain filter scheme. Only unlike the previous scheme, the information of multiple directions is used. This is one advantage of (7) and (10). But (10) has no the merits of the layered denoising. Inspired by (2) and (3), (10) is improved as

$$I_k(i, j) = I_{k-1}(i, j) + \frac{1}{C(i, j)} \sum_{(i', j') \in N_0(i, j)} e^{-\frac{\|I_{k-1}(N_s(i, j)) - I_{k-1}(N_s(i', j'))\|_{G_p}^2}{h^2}} (I_{k-1}(i', j') - I_{k-1}(i, j)) \quad (11)$$

Now, the scheme (11) is established. Gaussian filter is used. This is because that Gaussian filter achieves success in NLM. To remedy (11), two factors are introduced. One is h' , and it is a linear factor. Another is θ , and it is a nonlinear remedied factor.

The two factors are used to control the denoising amount at each step. So, the proposed scheme is formulated as:

$$I_k(i, j) = I_{k-1}(i, j) + \frac{h'}{C'(i, j) + \theta} \sum_{(i', j') \in N_0(i, j)} e^{-\frac{\|I_{k-1}(N_s(i', j')) - I_{k-1}(N_s(i, j))\|_{G_\rho}^2}{h^2}} (I_{k-1}(i', j') - I_{k-1}(i, j)) \quad (1 \leq k \leq K) \tag{12}$$

where $C'(i, j) = C(i, j) - 1$. The step size is $\frac{h'}{C'(i, j) + \theta}$. Obviously, the step sizes at the different locations are changed in a linear proportional fashion with h' . For so, h' is called a linear factor. On the other hand, $\frac{h'}{C'(i, j) + \theta} / \frac{h'}{C'(i, j)} = \frac{C'(i, j)}{C'(i, j) + \theta} = 1 - \frac{\theta}{C'(i, j) + \theta}$. Here, the parameter θ is a constant. Compared to NLM with “zero” center weight, the bigger $C'(i, j)$ is, the less the step sizes change in the proportion of the reduction. The smaller $C'(i, j)$ is, the greater the step sizes change in the proportion of the reduction. The noisy pixel with more similar pixels will be smoothed more deeply (at this time, $C'(i, j)$ is big). The noisy pixel with less similar pixels will be smoothed more slightly (at this time, $C'(i, j)$ is small). For so, θ is called a nonlinear remedied factor. It plays an important role in tuning the denoising amount at different locations.

In the classical NLM filter, θ is the weight of center pixel. It is taken to be 1 or 0 or the maximal value of weight of pixels in the neighborhood. In the different literature, the value of θ is presented in term of the different ways [12, 14, 17]. This means the role of θ is very important for the denoising performance. But to my knowledge, the optimal θ for the different images at the different cases is still in the study until now. The different literature presents the different computations of θ in NLM implementation. In the proposed diffusion iterative based denoising scheme, the value of θ will be set by empirically in the proposed iterative scheme. Good results are obtained. In the following Section 2.4, the determination of h', θ and other parameters will be detailed.

In (12), let

$$e^{-\frac{\|I_{k-1}(N_s(i', j')) - I_{k-1}(N_s(i, j))\|_{G_\rho}^2}{h^2}} (I_{k-1}(i', j') - I_{k-1}(i, j)) = threshold \cdot (I_{k-1}(i', j') - I_{k-1}(i, j)) \tag{13}$$

This shows that *threshold* is the function of the Gaussian filter result of square difference domain $(I_{k-1}(i'' + (i' - i), j'' + (j' - j)) - I_{k-1}(i'', j''))^2 = (\nabla I_{k-1}^{i', j'}(i, j))^2$ ((i'', j'') covers the image

domain) at (i, j) . And then, $threshold \cdot (I_{k-1}(i', j') - I_{k-1}(i, j))$ is rewritten as $e^{-\frac{(\nabla I_{k-1}^{i', j'})_{G_\rho}^2(i, j)}{h^2}} \cdot \nabla I_{k-1}^{i', j'}(i, j)$ in a simpler form where $(\nabla I_{k-1}^{i', j'})_{G_\rho}^2(i, j)$ represents the Gaussian filter result of squared difference gradient domain at (i, j) .

Finally, the proposed MGDID is expressed as follows:

$$I_k(i, j) = I_{k-1}(i, j) + \frac{h'}{C'(i, j) + \theta} \sum_{(i', j') \in N_0(i, j)} e^{-\frac{(\nabla_{i', j'}^{I_{k-1}})_{G_\rho}^2}{h^2}} \nabla I_{k-1}^{i', j'} (1 \leq k \leq K) \quad (14)$$

In fact, by contrasting (14) and (7), an important advantage of MGDID can easily be found. That is, the simplicity of the computation compared to the classical NLM (7). The denoising process of MGDID can be more easily implemented by matrix calculation. Since the optimization way of matrix calculation has been very mature, the computation time of the proposed scheme (14) can be reduced dramatically. Section 3 will verify this by running the related methods in the Matlab software.

2.4 Parameters for the proposed MGDID

In MGDID, the size of the filter windows including Gaussian kernel (similar window) and search window, standard deviation ρ of Gaussian kernel, h, h', θ and iteration number K need to be determined. An intuition is that algorithms should be equipped with almost same parameters for images with the similar structure. The idea has been successfully used in [20]. So, in practice, to determine the parameters under different cases for different images, one image can be taken as test image. This is because that each of natural images consists of many features such as homogeneous domains, edges and details and etc. The optimal parameters of other images can be roughly considered as the vicinity of optimal parameters of the taken image. Based on which, one can determine the parameters of other images by hand in the tests. In the test, the Lena image is taken to determine the parameters.

For each iteration step of the proposed method involves the contributions of more pixels compared to the traditional four-direction gradient domain, the less iteration number is needed. In the test, because of simplicity, $K = 2$ is selected. For this, the proposed algorithm is a two-stage algorithm in essence. The flowchart of algorithm is shown in Fig. 4. In the first stage (step), noise is removed roughly. In the second stage (step), noise is removed deeply and carefully.

In term of the above guidance of determining the parameters. to better preserve the edges, the sizes of the search and similar windows and the standard deviation of Gaussian kernel are smaller in the second step. Let the search window size and the similar window size vary in [3, 5, 7, 9, 11, 13, 15]. The standard deviation ρ of Gaussian kernel is in [1, 3, 5, 7, 9, 11, 13, 15]. The h' ranges in [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5]. In term of [20], the noise variance is $a\sigma^2$ ($a = 2$) in the gradient domain. In practice, a can be set be bigger so that noise can be removed better (in this test, a is taken as 2.2). Assume h is set to be h_1 in the first step, and then h is set to be $h_2 = \sqrt{a-h_1^2}\sigma$ in the second step. In this way, it is expected that

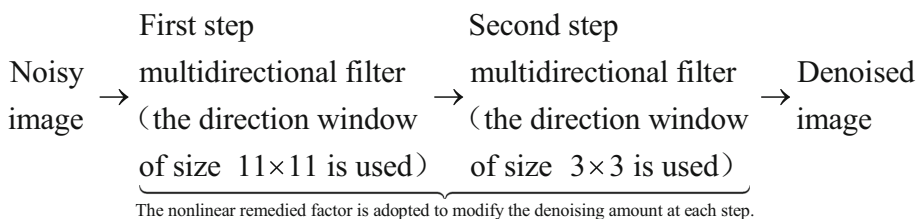


Fig. 4 The flowchart of the proposed MGDID

the noise can be completely eliminated. As for θ , it varies with $0.005\lambda(\lambda \in Z)$. It is noted that the other parameters of the first step are presented first by experience for some θ . And then, the other parameters of the second step is determined carefully when the high PSNR are get. And then, it is found that search window size, h' and h in two steps have been optimized relatively. Therefore, one adjusts θ by reducing similar window size and the standard deviations ρ of Gaussian kernel. The adjustment is repeated so that the good PSNR result is get.

After a series of tests, the parameters achieving the high peak signal-to-noise ratio (PSNR) is as follows. h_1 is 0.70σ and $h_2 = \sqrt{2.2-h_1^2}\sigma$ in turn. And then h' is 0.90 and 0.60 by hand in turn. Here, in the second step, the larger h will remove the noise located in image features such as edges deeply. In fact, the role of h has been displayed in Fig. 3. At the same time, smaller h' is used so that the noise can not be removed excessively. This is because that excessive removal of noise at feature points will destroy image information more seriously such as boundary, texture and so on if h is inappropriate but large. The search window sizes are 11 and 3 in turn. The similar window sizes (Gaussian kernel size) are 13 and 5 in turn. The standard deviations ρ of Gaussian kernel are 5 and 1 in turn from the first step to the second step.

And then, after other parameters is determined, for θ , it is found that the good results are obtained by taking $\alpha = 0.015$ with $\lambda = 3$. The influence of the parameter θ is shown in Fig. 5. In Fig. 5, the noisy Lena image ($\sigma = 20$) is processed. With the change of θ from 0 to 1, PSNR first increases and then decreases. The optimal θ is taken when the highest PSNR is obtained. The computational error changes the denoising contribution of different pixels so that the proper θ brings the good performance. The computation error behavior is very complex so that it is not possible to provide a more rigorous explanation for the successful use of θ and other parameters. Although other parameter sets are tried, the results with more gains are not obtained.

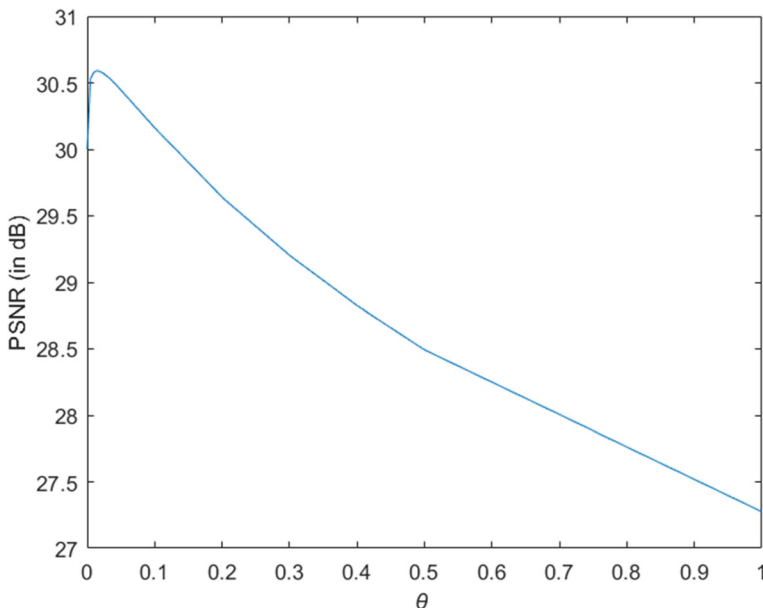


Fig. 5 The role of the nonlinear remedied factor θ for the denoising performance

3 The experimental results

The proposed method is applied to several test images corrupted with simulated noises at five different power levels $\sigma \in [5, 10, 15, 20, 25]$. The test set comprises images from the well-known images such as Lena, House, Boat, Barbara, Peppers and Cameraman of size 256×256 . Comparisons of restoration performance in terms of quantitative and visual comparisons have been made among the proposed MGDID, GRAD [2], NLM [1] (the center weight is taken as zero), NLM-SAP [3], NLM-LJSCPW [17] and LALWF [20]. For NLM-SAP, NLM-LJSCPW and LALWF, the parameters are set according to the value given by their authors in the corresponding paper. Only for LALWF, the iterative number (IN) is taken so that the highest PSNR is achieved. For GRAD, Gaussian kernel size and standard deviation are 3 and 1, respectively. The diffusion function is the exponential function proposed by Perona and Malik in [2]. The gradient threshold is 0.30σ . The iteration number is taken so that the best results are obtained. For NLM, the center weight is taken as zero. And other parameters are taken according to its Matlab code [10]. For the proposed method, the parameters are taken according to Section 2.4. Restoration performance is quantitatively evaluated by peak signal-to-noise ratio (PSNR) and mean structural similarity (MSSIM) [16] which are defined as

$$\text{PSNR(in dB)} = 10\log_{10} \frac{255^2}{\frac{1}{PQ} \sum_{i=1}^P \sum_{j=1}^Q (E_{i,j} - F_{i,j})^2} \quad (15)$$

$$\text{MSSIM}(E, F) = \frac{\sum_{ij=1}^R \text{SSIM}(x, y)}{R} \quad (16)$$

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\mu_x^2 + \mu_y^2 + C_2)} \quad (17)$$

respectively. In (15)–(17), PQ represents the image size; E and F denotes the original image and the denoised image, respectively. x and y are the image contents at the ij -th local window in the original and the denoised image, respectively. R is the total number of local windows in the image; μ_x and μ_y are the mean intensity of x and y ; σ_x and σ_y are the standard deviation; σ_{xy} is covariance between of x and y ; $C_1 = (k_1L)^2$ and $C_2 = (k_2L)^2$ are two variables. L is the range of the pixel values. The default k_1 and k_2 values are 0.01 and 0.03, respectively.

3.1 Comparison of PSNR and MSSIM values

Table 1 lists the PSNR and MSSIM values of three iterative methods including GRAD, LALWF and the proposed MGDID. The iteration number (IN) of three methods is also presented. The proposed method always outperforms the GRAD in PSNR and MSSIM. In most cases, the proposed MGDID also obtains the higher PSNR and MSSIM compared to LALWF. For example, under thirty cases in PSNR, the result with the proposed method is inferior to LALWF only in two cases. Under thirty cases in MSSIM, the result with the

Table 1 The iterative number (IN) used and the PSNR (in dB) and MSSIM results presented by GRAD, [2] LALWF [20] and the proposed method

Method	Metrics	Noise variance											
		25	100	225	400	625	25	100	225	400	625		
GRAD [2]	IN	Lena	3	21	26	29	30	House	12	29	36	39	42
	PSNR		35.78	31.97	30.05	28.81	27.85		36.83	34.17	32.73	31.66	30.81
	MSSIM		0.9334	0.8995	0.8689	0.8432	0.8195		0.9140	0.8752	0.8582	0.8451	0.8330
LALWF [20]	IN		19	16	15	14	17		16	16	16	15	15
	PSNR		37.49	33.62	31.51	30.03	28.86		38.27	35.01	33.46	32.13	31.00
	MSSIM		0.9545	0.9192	0.8887	0.8613	0.8322		0.9352	0.8834	0.8623	0.8415	0.8198
Proposed	IN		2	2	2	2	2		2	2	2	2	2
	PSNR		37.65	34.05	32.05	30.59	29.39		38.53	35.58	34.02	32.75	31.57
	MSSIM		0.9591	0.9264	0.8973	0.8677	0.8376		0.9394	0.8941	0.8706	0.8480	0.8231
GRAD [2]	IN	Boat	7	21	27	31	33	Barbara	2	6	10	13	16
	PSNR		36.26	32.42	30.43	29.12	28.16		35.27	30.63	28.23	26.66	25.54
	MSSIM		0.9493	0.9106	0.8737	0.8413	0.8148		0.9434	0.8828	0.8320	0.7851	0.7440
LALWF [20]	IN		2	16	15	15	14		16	16	14	13	13
	PSNR		33.71	33.78	31.61	30.10	28.95		37.00	32.64	30.34	28.73	27.53
	MSSIM		0.9611	0.9237	0.8861	0.8523	0.8266		0.9670	0.9304	0.8940	0.8538	0.8120
Proposed	IN		2	2	2	2	2		2	2	2	2	2
	PSNR		37.77	34.17	32.04	30.52	29.23		36.93	33.30	31.29	29.66	28.30
	MSSIM		0.9654	0.9292	0.8926	0.8585	0.8245		0.9694	0.9402	0.9057	0.8659	0.8247
GRAD [2]	IN	Peppers	14	27	30	32	33	Cameraman	2	14	22	26	30
	PSNR		36.22	32.85	31.00	29.71	28.73		35.93	31.58	29.65	28.37	27.40
	MSSIM		0.9395	0.9075	0.8854	0.8655	0.8480		0.9191	0.8885	0.8631	0.8356	0.8133
LALWF [20]	IN		19	16	16	15	15		16	16	16	15	16
	PSNR		37.63	34.06	32.05	30.52	29.31		37.89	33.46	30.99	29.40	28.33
	MSSIM		0.9493	0.9203	0.8975	0.8736	0.8512		0.9574	0.9218	0.8805	0.8419	0.8112
Proposed	IN		2	2	2	2	2		2	2	2	2	2
	PSNR		37.64	34.30	32.37	30.96	29.78		37.70	33.52	31.28	29.85	28.85
	MSSIM		0.9538	0.9243	0.8989	0.8733	0.8469		0.9589	0.9219	0.8794	0.8416	0.8116

Table 2 The PSNR (in dB) and MSSIM results presented by three NLM methods and the proposed method

Method	Metrics	Noise variance									
		25	100	225	400	625	25	100	225	400	625
NLM [1]	PSNR	31.93	31.77	30.83	29.68	28.63	House	34.77	33.44	32.08	30.80
	MSSIM	0.9322	0.9060	0.8725	0.8359	0.7985	36.25	0.8914	0.8585	0.8233	0.7850
	NLM-LJSCPW [17]	PSNR	37.19	33.21	30.91	29.30	28.08	38.44	35.18	31.69	30.39
		MSSIM	0.9547	0.9131	0.8745	0.8390	0.8078	0.9438	0.8952	0.8383	0.8135
	NLM-SAP [3]	PSNR	37.43	33.72	31.67	30.15	28.66	38.70	35.52	32.62	31.35
		MSSIM	0.9572	0.9208	0.8885	0.8581	0.8179	0.9444	0.8923	0.8495	0.8302
Proposed	PSNR	37.65	34.05	32.05	30.59	29.39	38.53	35.58	32.75	31.57	
	MSSIM	0.9591	0.9264	0.8973	0.8677	0.8376	0.9394	0.8941	0.8706	0.8480	
NLM [1]	Boat	PSNR	34.20	33.08	31.34	29.80	28.52	Barbara	32.64	30.95	29.32
		MSSIM	0.9508	0.9170	0.8749	0.8323	0.7952	33.31	0.8915	0.8457	0.7983
	NLM-LJSCPW [17]	PSNR	37.33	33.28	30.80	29.05	27.74	37.21	33.30	28.78	27.29
		MSSIM	0.9594	0.9123	0.8664	0.8250	0.7891	0.9686	0.9349	0.8929	0.8470
	NLM-SAP [3]	PSNR	37.53	33.85	31.72	30.05	28.63	36.84	33.45	29.65	28.21
		MSSIM	0.9641	0.9245	0.8830	0.8441	0.8062	0.9688	0.9403	0.9040	0.8607
Proposed	PSNR	37.77	34.17	32.04	30.52	29.23	36.93	33.30	29.66	28.30	
	MSSIM	0.9654	0.9292	0.8926	0.8585	0.8245	0.9694	0.9402	0.9057	0.8659	
NLM [1]	Peppers	PSNR	32.32	32.08	31.14	30.05	29.02	Cameraman	29.59	29.17	28.63
		MSSIM	0.9318	0.9080	0.8779	0.8445	0.8098	29.61	0.8906	0.8525	0.8137
	NLM-LJSCPW [17]	PSNR	36.96	33.27	31.05	29.43	28.11	37.12	33.07	29.35	28.16
		MSSIM	0.9466	0.9092	0.8771	0.8464	0.8162	0.9502	0.9147	0.8799	0.8465
	NLM-SAP [3]	PSNR	37.53	34.17	32.17	30.67	29.48	37.73	33.55	29.75	28.66
		MSSIM	0.9523	0.9203	0.8926	0.8662	0.8426	0.9579	0.9228	0.8815	0.8457
Proposed	PSNR	37.64	34.30	32.37	30.96	29.78	37.70	33.52	29.85	28.85	
	MSSIM	0.9538	0.9243	0.8989	0.8733	0.8469	0.9589	0.9219	0.8794	0.8416	

proposed method is inferior to LALWF only in five cases. Furthermore, the iteration number of the proposed method is fixed. The iteration number of LALWF has also some changes for different cases. That of GRAD has great changes for different cases. In practice, the iterative number is hard to manipulate. The proposed method avoids this problem. It is robust in PSNR and MSSIM, and can be implemented easily.

Table 2 lists the PSNR and MSSIM values of NLM, NLM-SAP, NLM-LJSCPW and the proposed method. Obviously, the MGDID method is more stable in terms of PSNR and MSSIM compared to other evaluated methods. It can be seen that the MGDID method is always better than NLM both in PSNR and MSSIM. For Lena, House and Peppers, the MGDID always obtains the highest PSNR and MSSIM. For House, Barbara and Cameraman, the proposed method does not always achieve the best results compared with NLM-SAP and NLM-LJSCPW. But the proposed MGDID is very close to the best results on average. NLM-LJSCPW has usually the bigger difference from the best results. For House, Barbara and Cameraman, the proposed method and NLM-SAP is very close in PSNR and MSSIM. But the proposed method always wins NLM-SAP for Lena, House and Peppers by bigger gains. For example, MGDID wins NLM-SAP 0.72 dB for Lena at $\sigma = 25$.

In a word, in terms of PSNR and MSSIM from Tables 1 and 2, the proposed method is stable and good in performance. Table 3 presents the comparison of LALWF, NLM and the proposed method. For Barbara image with the rich repetitive texture feature, the proposed MGDID outperforms NLM from 0.33 dB to 1.66 dB in PSNR. For Barbara image, NLM usually wins the local adaptive layered Wiener filter (LALWF) based on nonlinear diffusion Scheme 0.44 dB at least in PSNR. Only for Barbara at $\sigma = 5$, the proposed method is slightly inferior to LALWF in PSNR. In MSSIM, the proposed method almost always also outperforms LALWF and NLM under all kinds of cases. This shows that the proposed method is the successful development of NLM and nonlinear diffusion methods.

3.2 Comparison of visual performance

Figure 6 shows the denoised results of all the evaluated methods applied to the noisy Barbara image with $\sigma = 20$. The observation from Fig. 6 shows that the NLM and NLM-LJSCPW methods cannot suppress noise effectively. The NLM-SAP method leads to the over-smoothing effect near the edges and in the textural areas. In the result with the GRAD, there is a lot of speckles and noise. In the result with LALWF, there are the blurred textures and many spurious blocks. By comparison, the proposed MGDID method provides the best restoration results because it can suppress noise effectively while preserving image details no matter in smooth regions or detail regions. Figure 7 shows the denoised results of all the evaluated methods applied to the noisy Lena image with $\sigma = 10$. Compared to NLM and NLM-SAP methods, the proposed method better processes the details. For example, see the hair in the square domain in Fig. 7h. Compared to Fig. 7a, the details of hair are preserved well. But the details of hair in Fig. 7d with NLM and Fig. 7f with NLM-SAP are almost not invisible. The processed results with other methods produce some artifacts or have more residual noises. In a word, the proposed method better processes the noisy image from Figs. 6 and 7.

3.3 Comparison of computational time

The computational burden of the filters is measured as CPU time provided those are filtering the same image in Matlab on the same computer. Table 4 presents the computational time by

Table 3 The PSNR (in dB) and MSSIM results presented by NLM, [1] LALWF [20] and the proposed method

Method	Metrics	Noise variance											
		25	100	225	400	625	25	100	225	400	625		
NLM [1]	PSNR	31.93	31.77	30.83	29.68	28.63	House	34.77	33.44	32.08	30.80		
	MSSIM	0.9322	0.9060	0.8725	0.8359	0.7985	36.25	0.8914	0.8585	0.8233	0.7850		
	LALWF [20]	PSNR	37.49	33.62	31.51	30.03	28.86	38.27	35.01	33.46	32.13	31.00	
		MSSIM	0.9545	0.9192	0.8887	0.8613	0.8322	0.9352	0.8834	0.8623	0.8415	0.8198	
		Proposed	37.65	34.05	32.05	30.59	29.39	38.53	35.58	34.02	32.75	31.57	
NLM [1]	MSSIM	0.9591	0.9264	0.8973	0.8677	0.8376	0.9394	0.8941	0.8706	0.8480	0.8231		
	LALWF [20]	PSNR	34.20	33.08	31.34	29.80	28.52	Barbara	32.64	30.95	29.32	27.97	
		MSSIM	0.9508	0.9170	0.8749	0.8323	0.7952	33.31	0.9305	0.8915	0.8457	0.7983	
		Proposed	PSNR	33.71	33.78	31.61	30.10	28.95	0.9548	32.64	30.34	28.73	27.53
			MSSIM	0.9611	0.9237	0.8861	0.8523	0.8266	0.9670	0.9304	0.8940	0.8538	0.8120
Proposed			37.77	34.17	32.04	30.52	29.23	36.93	33.30	31.29	29.66	28.30	
NLM [1]	MSSIM	0.9654	0.9292	0.8926	0.8585	0.8245	0.9694	0.9402	0.9057	0.8659	0.8247		
	LALWF [20]	PSNR	32.32	32.08	31.14	30.05	29.02	Cameraman	29.59	29.17	28.63	27.99	
		MSSIM	0.9318	0.9080	0.8779	0.8445	0.8098	29.61	0.8906	0.8525	0.8137	0.7747	
		Proposed	PSNR	37.63	34.06	32.05	30.52	29.31	0.9196	33.46	30.99	29.40	28.33
			MSSIM	0.9493	0.9203	0.8975	0.8736	0.8512	0.9574	0.9218	0.8805	0.8419	0.8112
Proposed			37.64	34.30	32.37	30.96	29.78	37.70	33.52	31.28	29.85	28.85	
NLM [1]	MSSIM	0.9538	0.9243	0.8989	0.8733	0.8469	0.9589	0.9219	0.8794	0.8416	0.8116		

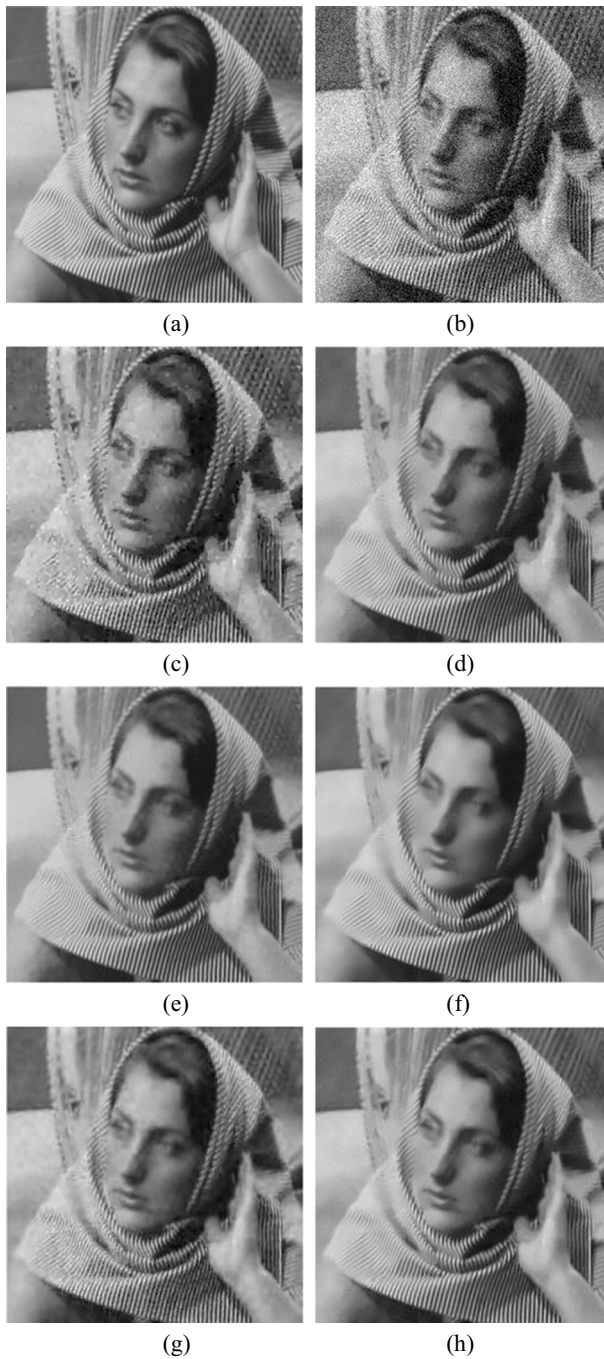


Fig. 6 Comparison of the restoration results from the different methods. Zoom into file for a better view. (a) original Barbara (256×256) image. (b) noisy image ($\sigma = 20$). (c-h) shows restored Barbara images using GRAD, NLM, NLM-LJSCPW, NLM-SAP, LALWF and Proposed, respectively



Fig. 7 Comparison of the restoration results from the different methods. Zoom into file for a better view. (a) original Lena (256×256) image. (b) noisy image ($\sigma = 10$). (c-h) shows restored Lena images using GRAD, NLM, NLM-LJSCPW, NLM-SAP, LALWF and Proposed, respectively

Table 4 The computation load by different methods

Method	GRAD [2]	NLM [1]	NLM-LJSCPW [17]	NLM-SAP [3]	LALWF [20]	Proposed
Computational time	1.27 s	45.28 s	21.71 s	131.54 s	0.87 s	2.42 s

different methods. The proposed method takes more times compared to GRAD and LALWF. However, the proposed method achieves the overall improvements in performance. Furthermore, the iteration number of the proposed method is fixed. The iteration number of GRAD and LALWF varies with images. More importantly, the proposed method saves more time compared to NLM, NLM-SAP, and NLM-LJSCPW when achieving good results. Therefore, the proposed MGDID is very efficient.

4 Conclusion

A new approach has been presented for performing high quality image denoising. It is the new development of LALWF. The proposed MGDID operates in multidirectional gradient domain. Inspired by NLM and LALWF, the parameters of the proposed MGDID are designed effectively. Experimental results demonstrate the proposed approach outperforms the state-of-the-art gradient domain and NLM methods in both objective and subjective quality. Especially compared to NLM methods, the proposed method is very efficient. Future research in this field includes optimization of the related parameter and selection of more effective pre-filter.

Acknowledgments This work is partially supported by National Natural Science Foundation of China (Grant No. 61401383), Basic Research Plan of Natural Science in Shaanxi Province (Grant No. 2021JM-518) and Qinglan Talent Program of Xianyang Normal University (Grant No. XSYQL201503).

References

1. Buades A, Coll B, Morel JM (2005) A non-local algorithm for image denoising. *IEEE Comput. Soc Conf. Comput. Vis Pattern Recognit* 2:60–65
2. Catté F, Lions P, Morel J (1992) Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J Numer Anal* 29(1):182–193
3. Deledalle CA, Duval V, Salmon J (2012) Non-local methods with shape-adaptive patches (NLM-SAP). *J Math Imaging Vis* 43(2):103–120
4. Eom IK, Kim YS (2004) Wavelet-based denoising with nearly arbitrarily shaped windows. *IEEE Signal Proces Lett* 11(12):937–940
5. Hjouji A, Jourhmane M, Jaouad EM, Es-Sabry M (2018) Mixed finite element approximation for bivariate Perona–Malik model arising in 2D and 3D image denoising. *3D Research* 9(3):36
6. Lai R, Yang Y (2011) Accelerating non-local means algorithm with random projection. *Electron Lett* 47(3): 182–183
7. Liao X, Yin J, Chen M, Qin Z (2020) Adaptive payload distribution in multiple images steganography based on image texture features. *IEEE T Depend Secure* (in press)
8. Liao X, Li K, Zhu X, Liu KJR (2020) Robust detection of image operator chain with two-stream convolutional neural network. *IEEE J-STSP* 14(5):955–968
9. Maiseli BJ (2020) On the convexification of the Perona–Malik diffusion model. *Signal Image Video P* 14: 1283–1291

10. Manjon-Herrera JV, Buades A (2008) Non-local means filter, Matlab code. Matlab Central File Exchange <http://www.mathworks.com/matlabcentral/fileexchange/13176-non-local-means-filter>.
11. Miheak MK, Kozintsev RK, Moulin P (1999) Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Signal Process Lett* 6(12):300–303
12. Nguyen MP, Chun SY (2017) Bounded self-weights estimation method for non-local means image denoising using minimax estimators. *IEEE Trans Image Process* 26(4):1637–1649
13. Perona P, Malik J (1990) Scale-space edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Mach Intell* 12(7):629–639
14. Salmon J (2010) On two parameters for denoising with non-local means. *IEEE Signal Process Lett* 17(3):269–272
15. Vignesh R, Oh BT, Kuo CC (2020) Fast non-local means (NLM) computation with probabilistic early termination. *IEEE Signal Process Lett* 17(3):277–280
16. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612
17. Wu Y, Tracey B, Natarajan P, Joseph P (2013) James-stein type center pixel weights for non-local means image denoising. *IEEE Signal Process Lett* 20(4):411–414
18. Zhang X (2016) Image denoising using dual-tree complex wavelet transform and wiener filter with modified thresholding. *J Sci Ind Res India* 75(11):687–690
19. Zhang X, Feng X (2014) Hybrid gradient-domain image denoising. *AEU-Int J Electron C* 68(3):179–185
20. Zhang X, Feng X (2015) Image denoising using local adaptive layered wiener filter in the gradient domain. *Multimed Tools Appl* 74(23):10495–10514
21. Zhang X, Feng X, Wang W, Zhang S, Dong Q (2013) Gradient-based wiener filter for image denoising. *Comput Electr Eng* 39(3):934–944

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Xiaobo Zhang received the M.S. degree in computational mathematics from Xidian University, Xi'an, China, in 2007, and the Ph.D. degree in applied mathematics from Xidian University, Xi'an, China, in March 2014, respectively. He is currently a professor with Xianyang Normal University, Xianyang, China. His research interests include wavelets and partial differential equations for image processing.