Multimedia Tools and Applications (2024) 83:7395–7416
https://doi.org/10.1007/s11042-023-15928-3

# A modified non-local means using bilateral thresholding for image denoising

Xiaobo Zhang[1]

Received: 7 December 2022 / Revised: 3 May 2023 / Accepted: 29 May 2023 /
Published online: 8 June 2023

**Abstract**
In non-local means (NLM) method, weight of each pixel in the neighbor centered on the reference noisy pixel plays a different role for performance of NLM for image denoising. The reference noisy pixel is called center pixel. Usually, weight of each pixel including center pixel in the neighbor (neighbor pixel) is computed based on the distance between neighbor pixel and center pixel. This paper proposes a novel weight by studying the recent statistical nearest neighbors distance measurement (SNNDM) and gradient domain filter. The difference of each neighbor pixel including center pixel is considered sufficiently. The proposed weight is called the bilateral thresholding since it is similar to bilateral filtering in form. Test results show that the proposed method can deal with each neighbor pixel differently so that the desired performance is achieved.

**Keywords** Image denoising · Non-local means (NLM) · Statistical nearest neighbors distance measurement (SNNDM) · Bilateral thresholding · Gradient domain filter

## 1 Introduction

In the Multimedia era, denoising becomes a key challenge in the field of image processing [11]. A good denoised image can pave a way to many subsequent processing tasks such as image segmentation, object recognition and so on. For example, in [12], image denoising method was used for image recognition. The goal of image denoising is to reconstruct the latent clean image from its noisy counterpart while preserving the fine structures, textures and details as accurately as possible. Currently, a wide variety of strategies including total variation [10, 19] have been proposed to remove the noise from the degraded image. Especially, non-local means (NLM) proposed by Buades et al. in 2005 [1] has always been popular. The denoising mechanism of NLM is: denoising of a given pixel is obtained as a weighted average of the surrounding pixels, with weights proportional to the patch similarity measurement. In NLM, patch similarity is called non-local self-similarity (NLSS).

✉ Xiaobo Zhang
zhangxiaobo419@126.com

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

The emergence of NLSS has triggered a revolutionary change in the field of image processing. A number of the state-of-the-art methods armed with NLSS have emerged. For example, BM3D was proposed in [2]. This method obtains good results by grouping of similar image and specially developed collaborative Wiener filtering. BM3D is a milestone work employing a NLSS for grayscale image denoising. Another celebrated evolved version of NLM is the weighted nuclear norm minimization (WNNM) method in [8]. The method represented NLSS by using the weighted nuclear norm of similar patch matrices. Compared to BM3D, WNNM achieves better results with large performance gain. To better restore the structure and the edge of image contaminated by noise, a novel multi-scale weighted group sparse coding model (MS-WGSC) was proposed for image denoising by simultaneously exploiting the NLSS prior from the original scale and a coarse scale in [17]. In the recent years, NLSS is also incorporated into the deep learning architecture so that more powerful denoising performance is shown. For example, in [13, 24], the NLSS denoising algorithms based on deep learning techniques were used to successfully deal with the different noisy image. Although these methods have achieved good performance, they are becoming increasingly complex. So, some scholars have been paying attention to NLM itself, optimizing and improving the original NLM. This is also the motivation for the proposed method.

## 1.1 Related work

Currently, there are also some important studies in which original NLM filter is discussed deeply. In these researches, the weighted average mechanism of NLM filter is further revealed and optimized. These studies have to some extent overcome the shortcomings of NLM, such as high computational complexity, difficulty in determining parameter values, and unreasonable center weight values. For example, the literature [7, 22] studied how to accelerate NLM to make it useful in practical application. The literature [16, 18, 21, 26–29, 31] and [3, 9] concentrated their attention on how to determine the smoothing parameter, how to compute the weight of the center pixel (center weight) and how to select the weight function, respectively. These studies further improve NLM from different perspectives. This paper focuses on the distance between the pixels so that the proposed weight computes the similarity between the pixels properly.

In [5], Alessandro Foi et al. installed the foveated self-similarity into NLM (NLM-Fov). Their studies show that the foveation operators producing the best denoising performance are in complete line with the directional preference of the human visual system. In [4], Vadim Fedorov et al. exploited the affine invariant self-similarities to design a new NLM filter. The similarity measure compares patches related by a local affinity and considers shape adaptive patches so that a better denoised result is provided. In [6], the statistical nearest neighbors distance measurement (SNNDM) was introduced into the NLM filter (NLM-SNN) for measuring the distance between two patches so that an improvement is obtained in the perceived image quality. In [14], the pixel-pixel unbiased distance, patch-patch unbiased distance and coupled unbiased distance were used to construct the unbiased distance based non-local adaptive means. The tests in [14] show that a better performance than the state-of-the art denoising methods can be achieved by this method. All these methods tap the potential of different similarity measurement ways from different views. But, the methods in [4, 5] have a little complex. The method in [14] combines a few unbiased distance so that there is more complex. In contrast, the method in [6] is simple and effective. In addition, in [5, 14], NLM is carried out with the pixel wise scheme. In [4, 6], the patch wise scheme of NLM is implemented. By comparison, the pixel wise implementation is simpler and more convenient. So, starting from the SNNDM proposed in [6], this paper will propose a novel distance similarity measurement method for the pixel wise NLM by a simple design.

It is noted that the proposed method is different from the recent closely related pixel wise NLM methods [27–29]. In [27], the center weight (CW) was researched carefully and then a wiener filter center weight (WFCW) was proposed. This NLM method is called NLM-WFCW. In [28], a multidirectional gradient domain nonlinear diffusion image denoising scheme (MGDID) was proposed by introducing the remedy reactor of the step size from the view of NLM. In essence, MGDID presented a novel center weight in iterative implementation of NLM. In [29], a two-step iterative NLM scheme (TSNLM) was proposed by considering the role of the different center weights. The disadvantage of these NLM methods is that they only consider the role of center weight. Furthermore, more NLM filters are used to design MGDID in [28] and TSNLM in [29]. This increases the complexity of algorithm design and computation. The proposed method is like the original NLM in [1] and WFCW in [27], using only one NLM. The novelty of the proposed method lies in designing a new similarity measurement method that distinguishes and calculates the weights of different pixels.

The main contribution of the proposed method is in: (1) the statistical nearest neighbors distance measurement (SNNDM) is analyzed carefully; (2) to sufficiently consider the role of different neighbor pixel including the center pixel, the geometric distance is introduced into the computations of the weights of all neighbor pixels including the center pixel around the reference noisy pixel; (3) the relation between the parameters is discussed deeply; and (4) the tests show that the proposed method is effective by providing desired peak signal-to-ratio (PSNR), structural similarity (SSIM) and good visual effect. Furthermore, the proposed method is very simple.

The paper is organized as follows. In Section 2, one provides a detailed description of the proposed method. Then, in Section 3, the experimental results and comparisons are presented to demonstrate the effectiveness of the proposed method. Finally, one concludes this paper in Section 4.

## 2 The proposed method

In this section, the several aspects of the proposed method are detailed in turn. NLM filter is firstly reviewed. Then, the statistical nearest neighbors distance measurement (SNNDM) is introduced and analyzed. After that, bilateral thresholding is proposed to compute the weight of different pixel. At last, the parameters are discussed carefully.

### 2.1 Non-local means filter

For any pixel at $(x, y)$ in the image domain $\Omega$, the observation model has the following formulation:

$$I(x, y) = s(x, y) + \eta(x, y) \tag{1}$$

where $I(x, y)$ denotes noisy image, $s(x, y)$represents original noise free image, and $\eta(x, y)$ is independent and the zero-mean white Gaussian noise of variance $\sigma^2$. For any pixel of interest $(x, y)$, an estimate $u(x, y)$ with the NLM filter is given by:

$$u(x,y) = \frac{\sum_{(x',y')\in \mathrm{R}} I\left(x',y'\right)\omega\left(x,y,x',y'\right)}{\sum_{(x',y')\in \mathrm{R}} \omega(x,y,x',y')} \tag{2}$$

where R is a search window around $(x,y)$ and $\omega\left(x,y,x',y'\right)$ is the weights that compare the neighborhood patches $P(x,y)$ and $P\left(x',y'\right)$ around $(x,y)$ and $\left(x',y'\right)$, respectively. The original weights are defined as:

$$\omega\left(x,y,x',y'\right) = \exp\left(-\frac{\|I(P(x,y)) - I(P(x',y'))\|_{2,\rho}^2}{h^2}\right) \tag{3}$$

where $h > 0$ is the smoothing parameter controlling the filtering degree. $\left\|I(P(x,y)) - I\left(P\left(x',y'\right)\right)\right\|_{2,\rho}^2$ is a weighted Euclidean distance between two image patches $P(x,y)$ and $P\left(x',\ y'\right)$ of size $\left(2L_p+1\right)\times\left(2L_p+1\right)\left(L_p \geq 0 \ is\ a\ integer\right)$ defined as

$$d_G^2\left(x,y,x',y'\right) = \|I(P(x,y)) - I(P(x',y'))\|_{2,\rho}^2 = \sum_{i=-L_p}^{L_p}\sum_{j=-L_p}^{L_p} G_\rho(i,j)\left(I\left(x'+i,y'+j\right) - I(x+i,y+j)\right)^2 \tag{4}$$

where $G_\rho(i,j) = \frac{1}{\sqrt{2\pi}\rho}\exp\left(\frac{i^2+j^2}{2\rho^2}\right)$ is the Gaussian kernel with the smoothing parameter $\rho > 0$, and the $(i,j)$ varies in the similar window R′. Some alternative strategies have been proposed to measure the distance between two different pixels to obtain the high image processing quality [4–6, 14]. In term of (3), the computation of the distance between the same two pixels $I(x,y)$ will produce the center weight (CW) $\omega(x,y,x,y)$, and CW is always 1. This is unreasonable in practice. So, the CW problem has been discussed in [16, 18, 26–29] and some more effective CWs have been presented. In this paper, the proposed weight includes the computation of CW and all weights of other neighboring pixels. It is noted that (2) is expression of pixel wise NLM filter. (2) is turned into patch wise NLM filter when $I\left(x',y'\right)$ is replaced with $I\left(P\left(x',y'\right)\right)$. At this time, apart from the aggregation step, the estimate $u(P(x,y))$ of the noise-free patch of noisy patch is obtained. In the patch wise implementations of NLM filter, the weight of the patch $I\left(P\left(x',y'\right)\right)$ is computed as $\omega\left(x,y,x',y'\right)$ in term of the pixel wise implementation. That is that computation of weight of pixel $I\left(x',y'\right)$ in the pixel wise NLM is used as computation of weight of patch $I\left(P\left(x',y'\right)\right)$ in the patch wise NLM.

## 2.2 Statistical nearest neighbors distance measurement (SNNDM)

If the reference patch $P(x,y)$ and its neighbor $P\left(x',y'\right)$ are two noisy replicas of the same clean patch, one gets the expectation of the distance.

$$E\left[d^2\left(x,y,x',y'\right)\right] = 2\sigma^2 \tag{5}$$

where.

$$d^2\left(x,y,x',y'\right) = \|I(P(x,y)) - I(P(x',y'))\|_2^2 = \frac{1}{\left(2L_p+1\right)^2}\sum_{i=-L_p}^{L_p}\sum_{j=-L_p}^{L_p}\left(I\left(x'+i,y'+j\right) - I(x+i,y+j)\right)^2 \tag{6}$$

The above (6) is called the nearest distance. From (5), one can draw a conclusion that the expected distance is not zero for two noisy replicas of the same patch. Therefore, for patch wise NLM filter, unlike traditional distance measure method between two patches, the statistical nearest neighbors distance measurement (SNNDM) is adopted in [6]. According to SNNDM, one computes the similarity between patches whose distance from the reference patch is instead closed to its expectation. The SNNDM between two patches $P\left(x', y'\right)$ and $P(x, y)$ is defined as:

$$d^2_{SNNDM}\left(x, y, x', y'\right) = \max\left(d^2\left(x, y, x', y'\right) - 2\sigma^2, 0\right) \tag{7}$$

In (7), the $P\left(x',\ y'\right)$ must satisfy:

$$d^2_{SNN}\left(x, y, x', y'\right) = \left| d^2\left(x, y, , x', y'\right) - 2\widetilde{o}\sigma^2 \right| < \tau \tag{8}$$

where $\widetilde{o}$ is 1 for the SNNDM approach. Here, $\tau \geq 0$is a threshold on $P\left(x', y'\right)$. $\tau$ensures that the enough similar patches can be selected for every $P(x, y)$. The traditional Euclidean nearest neighbors distance measurement is a special case of (7) with $d^2_{SNN}\left(x, y, x', y'\right)$ when $\widetilde{o} = 0$. In [6], the SNNDM is used to improve the patch wise NLM filter. Compared to the nearest neighbors distance measurement (NNDM) NLM filter, the better denoising effect is achieved under the case that the fewer neighbor patches in search window are used. The details can be found in [6]. Inspired by [6], this paper will develop the weight function based on the distance measurement in the pixel wise NLM filter (2).

### 2.3 Bilateral thresholding

The proposed bilateral thresholding weight function is produced from three perspectives. The rationality of SNNDM will be incorporated into the new design of the weight function for the pixel wise NLM.

As mentioned earlier, SNNDM is produced by discussing two noisy replicas of the same clean patch. On the contrary case, NNDM is proper for two noisy image patches containing the same noise patch. At this time, $d^2\left(x, y, x', y'\right)$ will embody the ideal distance between $P(x, y)$ and , but SNNDM will bring bigger distance error because some corresponding patches may have been abandoned. For example, for the case that the $d^2_{SNN}\left(x, y, x', y'\right) > \tau$ but also $d^2\left(x, y, x', y'\right) < -\tau + \widetilde{o} \cdot 2\sigma^2$, one should pay more attention since as mentioned above that the pixel with smaller $d^2\left(x, y, x', y'\right)$ maybe has more contributions for NLM filter. In fact, the weight of the unselected (abandoned) patch is zero for SNNDM and NNDM for the patch wise NLM filter. This irrationality should not appear in the proposed pixel wise NLM filter. All neighborhood pixels should contribute to denoising. The roles of different pixels should be reflected by different weights. This is the fist perspective producing the proposed weight.

In fact, after a series of manipulations, the non-local scheme (2) can be re expressed as:

$$u(x, y) = I(x, y) + \frac{1}{C(x, y)} \sum_{(x', y') \in N(x, y)} e^{-\frac{\| I(P(x, y)) - I(P(x', y')) \|^2_{2,\rho}}{h^2}} \left( I\left(x', y'\right) - I(x, y) \right) \tag{9}$$

where $C(x, y) = \sum_{(x', y') \in \mathrm{R}} \omega\left(x, y, x', y'\right)$. It can be seen that (9) can be considered as one step multidirection gradient domain filter scheme. In (9), the information of multiple directions is used. The multidirection window is shown in Fig. 1. The multidirectional gradient domain image denoising (MGDID) presented the good denoised results [28]. Let represents the differences of different distances in the different

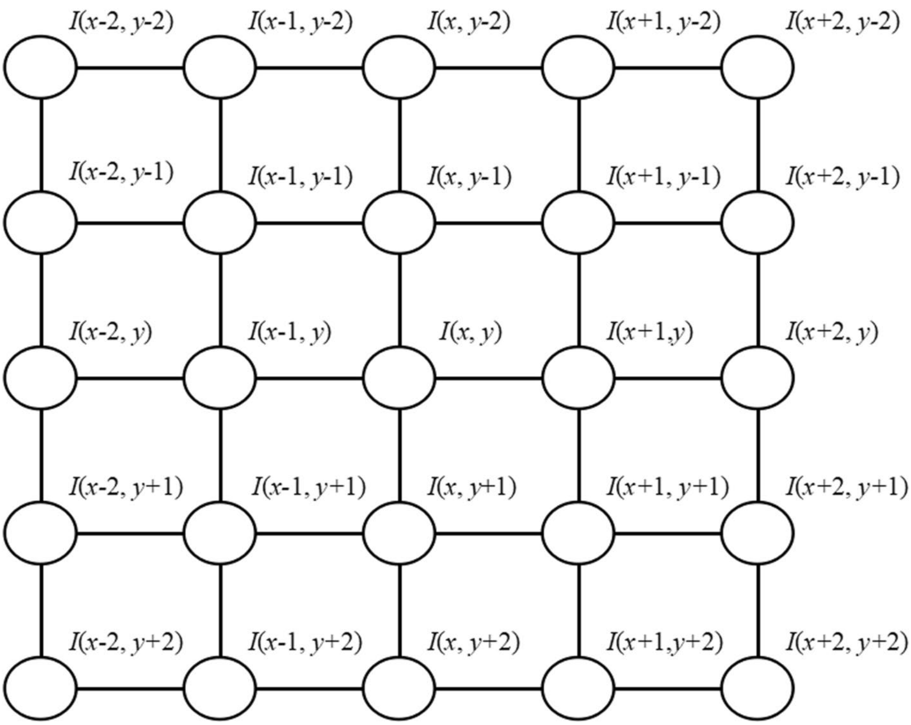

**Fig. 1** The "square" search window

direction $(\Delta x, \Delta y)$ centered at $(x, y)$ in the "square" search window, respectively. $\nabla I^{(\Delta x, \Delta y)}(x, y)$ is called gradient coefficient at $(x, y)$ with directions $(\Delta x, \Delta y)$. In [30], one implemented the local adaptive Wiener filter in the gradient domain because the gradient coefficients of image can be modeled as zero-mean Gaussian random variables with high local correlation. The gradient coefficient variance is estimated by using an approximate maximum likelihood (ML). So, the "signal" variance of the gradient coefficient of the latent clean image is expressed as:

$$\nabla I^{(\Delta x,\Delta y)}(x, y) = I(x + \Delta x, y + \Delta y) - I(x, y) = I\left(x', y'\right) - I(x, y) \tag{10}$$

$$d^2_{SVDM}\left(x, y, x', y'\right) = \max\left(q(x, y) - 2o\sigma^2, 0\right) \tag{11}$$

where $o \geq 0$, and the $q(x,y)$ in the size $(2L_p+1)\times(2L_p+1)$ local window is computed as $d^2\left(x, y, x', y'\right)$ by (6). Here, (11) is called the "signal" variance distance measurement (SVDM). From NLM filtering (2) and gradient domain filtering (9), one can draw a conclusion $d^2_{\boldsymbol{SVDM}}\left(x, y, x', y'\right)$ is another statistical distance measurement. The $d^2_{\boldsymbol{SVDM}}\left(x, y, x', y'\right)$ is zero when $q(x,y) \leq 2o\sigma^2$ for patch $P\left(x', y'\right)$ and $P(x,y)$. But, in SNNDM, $d^2_{\boldsymbol{SNNDM}}\left(x, y, x', y'\right)$ is considered infinity when $d^2_{SNN}\left(x, y, x', y'\right) = \left|d^2\left(x, y, x', y'\right) - 2\tilde{o}\sigma^2\right| > \tau$ for patch $P\left(x', y'\right)$ and $P(x,y)$. This means that the patch (pixel) with $d^2_{SNN}\left(x, y, x', y'\right) > \tau$ will be abandoned in the process of denoising .

The difference with SNNDM (7) and SVDM (11) lies mainly in that the two measurements have different expressions for $d_{SNN}^{2}\left(x,y,x',y'\right) > \tau$. In fact, both two different expressions have some disadvantages. For SNNDM, the disadvantage is that the unselected patch (pixel) is not considered as mentioned earlier. For SVDM, the disadvantage is also obvious. That is that the distance between two clean patches should all not be always set to zero when $q(x,y) \leq 2o\sigma^2$. In practice, two clean patches have difference on average. This is the second perspective producing the proposed weight.

The above analysis gave birth to a new weight based on the distance measurement. It is noted that if $\omega\left(x,y,x',y'\right)$ in (2) is turned into:

$$\omega_{bi}\left(x,y,x',y'\right) = \exp\left(-\frac{\left(x-x'\right)^2+\left(y-y'\right)^2}{h_s^2}\right)\exp\left(-\frac{\left(I(x,y)-I\left(x',y'\right)\right)^2}{h^2}\right) \tag{12}$$

and then NLM filter is changed into bilateral filtering in [20]. In (12), $h_s^2$ and $h^2$ are two different smoothing parameters used in spatial and range weights. $h^2$ in bilateral filtering plays the same role as that in NLM filter. Compared with bilateral filtering (12), NLM only computes the range weight. In addition, the range distance between two pixels takes into account two image patches instead of two single pixels. In NLM filter, the spatial distance is discarded. In the proposed method, the role of the spatial distance will be brought into full play. This is the third perspective producing the proposed weight. The pixel close to the center pixel in the spatial distance will have greater weight, and which has achieve success in bilateral filtering.

The proposed weight function adopts the following form:

$$\omega_{pr}\left(x,y,x',y'\right) = \underbrace{\exp\left(-\frac{\left(x-x'\right)^2+\left(y-y'\right)^2}{h_s^2}\right)}_{\phi_s(x,y,x',y')}\underbrace{\exp\left(-\frac{d_{pr}^2\left(x,y,x',y'\right)}{h^2}\right)}_{\phi_r(x,y,x',y')} \tag{13}$$

where.

$$d_{pr}^2\left(x,y,x',y'\right) = \max\left(\left\|I(P(x,y)) - I(P(x',y'))\right\|_{2,\rho}^2 - o_{pr}\sigma^2, 0\right) \tag{14}$$

From (13), the proposed distance measurement is as follow as:

$$d_{PRDM}^2\left(x,y,x',y'\right) = -\left[\ln\phi_s\left(x,y,x',y'\right) + \ln\phi_r\left(x,y,x',y'\right)\right] \tag{15}$$

Here, for the sake of the convenience, the smooth parameter $h_s^2$ and $h^2$ are incorporated into (15). Compared to the traditional NLM filter, the proposed distance measurement includes both spatial distance measurement and range distance measurement based on the pixel intensities. Besides this, compared to the "signal" variance distance measurement $d_{\boldsymbol{SVDM}}^2\left(x,y,x',y'\right)$, a key difference lies in that the Gaussian weighted Euclidean distance between two image patches is used. Compared to the SNNDM $d_{\boldsymbol{SNNDM}}^2\left(x,y,x',y'\right)$ with (7), the proposed distance aims to employ the difference of the different pixels in the neighbor of the pixel at $(x,y)$ for the pixel wise NLM filter. And all the neighbors pixels are used. The patch wise NLM filter with (7) is a two-step algorithm in essence. The first step is the weighted computation based on the similar patches. The second step is the aggregation of multiple estimates of the reference noisy pixel. The proposed pixel wise NLM filter is one step weighted method. The proposed method employing all neighbor pixels will demonstrate the satisfied performance.

Currently, the remained problem is the determination of the weight of the center pixel (CW) at $(x, y)$. The proposed weight $\omega_{pr}\left(x, y, x', y'\right)$ is product of the spatial kernel $\phi_s\left(x, y, x', y'\right)$ . For the CW $\omega_{pr}(x, y, x, y)$, both $\phi_s(x, y, x, y)$ and $\phi_r(x, y, x, y)$ are 1 in term of (13). As a matter of fact, the CW plays an important role for the performance of NLM filter. So, some optimal weights are presented in the recent years. In [27], the Wiener filter center weight (WFCW) achieves the very good effect. For the proposed method, the proposed center pixel distance is.

$$\overline{d}^{2}_{pr}(x, y, x, y) = \frac{1}{|\mathrm{R}_0|} \sum_{(x',y')\in \mathrm{R}_0} \left\| I(P(x, y)) - I(P(x', y')) \right\|^2_{2,\rho} \tag{16}$$

where $\mathrm{R}_0 = \mathrm{R} - (x, y)$. The signal variance at $(x, y)$ is computed as:

$$d^2_{pr}(x, y, x, y) = \max\left(\overline{d}^{2}_{pr}(x, y, x, y) - v\sigma^2, 0\right) \tag{17}$$

At last, the range kernel of the proposed CW is expressed as:

$$\phi_r(x, y, x, y) = \frac{v\sigma^2}{d^2_{pr}(x, y, x, y) + v\sigma^2} \tag{18}$$

The constant $v$ is used to tune the denoising extent in local Wiener filter $\phi_r(x, y, x, y)$. As with [27], (18) shows $\phi_r(x, y, x, y)$ is larger in the flat domain, and smaller in the edge and other characteristic areas. This will be conducive to the preservation of image features in the process of denoising.

The proposed complete weight function is produced as:

$$\omega_{cpr}\left(x, y, x', y'\right) = \begin{cases} \underbrace{\exp\left(-\frac{\left(x-x'\right)^2+\left(y-y'\right)^2}{h_s^2}\right)}_{\phi_s(x,y,x',y')} \underbrace{\exp\left(-\frac{d^2_{pr}\left(x, y, x', y'\right)}{h^2}\right)}_{\phi_r(x,y,x',y')} & \text{for } (x, y) \neq \left(x', y'\right) \\ \frac{v\sigma^2}{d^2_{pr}(x,y,x,y)+v\sigma^2} & \text{for } (x, y) = \left(x', y'\right) \end{cases} \tag{19}$$

And then, the restored image with the proposed method is computed as:

$$u(x, y) = \frac{\sum\limits_{(x',y')\in \mathrm{R}} I\left(x', y'\right)\omega_{cpr}\left(x, y, x', y'\right)}{\sum\limits_{(x',y')\in \mathrm{R}} \omega_{cpr}(x, y, x', y')} \tag{20}$$

The flowchart of the proposed method is shown in Fig. 2. It clearly and concisely indicates that the key in the proposed method lies in the calculation of the weight values in term of (19).

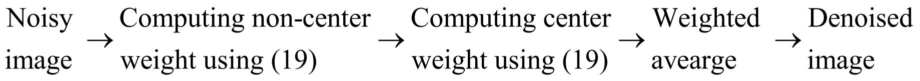
Noisy image
Computing non-center weight using (19)
Computing center weight using (19)
Weighted aveage
Denoised image

**Fig. 2** The flowchart of the proposed modified NLM method

### 2.4 Parameters

In the test, one takes one image as test image. The optimal parameters by hand can be obtained for the test image. And then, the optimal parameters of other images can be roughly considered to be around the optimal parameters of the test image. This is because each natural image is composed of similar features such as smooth domains, boundaries, textures, and details and etc. Thus, the parameters of other images can be roughly obtained. This strategy has been successfully used in the gradient domain methods in [28, 30].

Many parameters are involved in the proposed method. They include the size of search widow and similar window, the smoothing parameter $h_s^2$ of $\phi_s\left(x, y, x', y'\right)$ and $h^2$ of $\phi_r\left(x, y, x', y'\right)$, the offset parameter $o_{pr}$ in (14), the smoothing parameter $\rho$ of Gaussian kernel, etc.

For the sake of fair comparison, the size of search window and similar window is set to $11 \times 11$ which is the same as selection of [27]. Then selection of Gaussian kernel is considered. If the smoothing parameter ρ of Gaussian kernel is too small, it cannot achieve the proper filtering effect, and if the smoothing parameter ρ of Gaussian kernel is too large, it will blur the image features. After a lot of attempts, the smoothing parameter ρ of Gaussian kernel is taken as 2 by hand. Based on which, Gaussian kernel $G_\rho$ is produced by the MATLAB file "fspecial". In the test, it is found that the difference between results obtained by using the proposed Gaussian kernel and results obtained by Gaussian kernel used by Manjon Herrera and Buades is subtle for the proposed modified NLM method, and due to fair comparison, the Gaussian kernel used by Manjon Herrera and Buades in [15] is adopted in the proposed modified NLM. Gaussian kernel used by Manjon Herrera and Buades avoids selection of the smoothing parameter $\rho$ of Gaussian kernel. The details can be found in [15].

As for $v$ in (19), it is taken to 2. This is because that $\overline{d}_{pr}^2(x, y, x, y)$ is average of noisy gradient coefficient of different direction and different distance gradient domain in neighbor around $(x, y)$. From [30], the variance of Gaussian noise component contained in every noisy gradient coefficient is $2\sigma^2$. For so, 2 is a reasonable selection.

How to determine the value of the smoothing parameter $h_s^2$, $h^2$ and the offset parameter $o_{pr}$? Let $h_s^2 = o_s h_{s0}$ and $h^2 = o_r h_{r0}$($h_{s0}$ and $h_{r0}$ are the initial value). Here, the offset parameter $o_s$ and $o_r$ is used to tune the corresponding parameter value. And then, one presents a simple fitted approach to describe $o_{pr}$, $o_s$ and $o_r$. Assume that these offset parameters vary linearly with noise level. The function model is as follows:

$$f = f_0 + \theta\left(\sigma - \sigma_0\right) \quad \left(f \in \left\{o_{pr}, o_s, o_r\right\}\right) \tag{21}$$

where $f_0$ is preset optimal value under some noise level $\sigma_0$ for some image. The (21) is used to get the corresponding functions of different offset parameter $o_{pr}$, $o_s$ and $o_r$.

Here, the peppers image is taken to obtain the initial values and optimal values. Let $\sigma_0 = 10$, under this noise level, the corresponding parameters are tuned. For example, $h_{s0}$ and $h_{r0}$ are 40 and $\sigma^2$, respectively. For $o_{pr}$, $f_0 = o_{pr0} = 1$. For $o_s$, $f_0 = o_{s0} = 1$. For $o_r$, $f_0 = o_{r0} = 1$. To determine the $\theta$, let $\sigma = 25$. The corresponding $o_{pr}$, $o_s$ and $o_r$ can be presented. And then, the different $\theta$ is determined. For $o_{pr}$, $\theta = 0.08$. For $o_s$, $\theta = -0.02$. For $o_r$, $\theta = -0.02$. Three functions are shown in the following, respectively.

$$o_{pr} = 1 + 0.08(\sigma - 10) \tag{22}$$

$$o_s = 1 - 0.02(\sigma - 10) \tag{23}$$

$$o_r = 1 - 0.02(\sigma - 10) \tag{24}$$

The function (22) demonstrates that $o_{pr}$ increases as $\sigma$ increases. At this time, $d_{pr}^2(x,y,x',y')$ decreases. This means more $d_{pr}^2(x,y,x',y')$ is zero. More pixels around $(x,y)$ will have the same value 1 of $\phi_r(x,y,x',y')$ for higher noise level. At this time, the proper $o_s$ and $o_r$ will tune the denoising contribution of these pixels with the same $d_{pr}^2$. The differences of pixels will be fully utilized. Some important notations in the proposed method are summarized in Table 1.

## 3 Experimental results

The aim of this section is to validate the performance of the proposed method. The comparison results are obtained by some NLM based methods using a NLM filter. They include the pixel wise methods using a NLM filter such as NLM-Stein [18], NLM-LJS [26], NLM-Max [15], NLM-WFCW [27] and NLM-Fov [5] since the proposed method is a kind of pixel wise design using a NLM filer. Among these methods, NLM-Fov is a state-of-the-art method for NLM filter. To further verify the performance of the proposed method and also for the proposed method is variant of NLM-SNN, the patch wise NLM filter NLM-SNN [6] is also used to compare. All the simulations are performed under the MATLAB R2017a environment. The code of the proposed NLM comes from NLM-Max [15] and can also originate from NLM-LJS [25]. The performance is similar. Therefore, the result is produced by modified code only from [15]. The main difference lies in that the running time is different. The 10 grayscale test images including Cameraman (256×256), Lena (512×512), Montage (256×256), House (256×256), Pepper (256×256), Barbara (512×512), Boat (512×512), Hill (512×512), Couple (512×512) and Fingerprint (512×512) are used to evaluate subjective and objective performance of the different NLM filter.

For the proposed method, the parameters have been determined in Section 2.4. For NLM-Stein, NLM-LJS and NLM-WFCW, as with [27], for the smoothing parameter $h$, the value is $\sigma$. The search window and similar window is the same as that of the proposed method. The used Gaussian kernel is that used by Manjon Herrera and Buades in [15]. For NLM-Fov and NLM-SNN, the parameters of each method have been set according to the values given by their respective authors in the corresponding referred codes. Only for NLM-SNN, $h$is taken to $\sigma$ so that the better result is get.

**Table 1** Some important notations for the proposed method

| Notation | Meaning |
|---|---|
| $o_{pr}$ | It is the offset parameter in (14). |
| $o_s$ | It is the offset parameter in $h_s^2 = o_s h_{s0}$ in the spatial kernel. |
| $o_r$ | It is the offset parameter in $h^2 = o_r h_{r0}$ in the range kernel. |
| $\theta$ | It is the parameter in the proposed parameter tuned model (21). It varies with $o_{pr}$, $o_s$ and $o_r$. |
| $d_{pr}^2$ | It is the distance expression in the proposed range kernel between pixels. |
| $\phi_r$ | It is the proposed range kernel. |
| $\phi_s$ | It is the proposed spatial kernel. |
| $\omega_{pr}$ | It is the proposed weight. |
| $v$ | It is concerned with the computation of CW in (19). It is set to 2. |

The denoising performance is measured in term of the peak signal-to-ratio (PSNR) and structural similarity (SSIM) index. PSNR measures the intensity similarity between the noise-free image $s$ and the denoised image $u$. It is defined as.

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{\text{MSE}}\right)\text{dB} \tag{25}$$

where MSEis the mean squared error between the original noise-free image $s$ and the reconstructed $u$. SSIM [23] can understand the structure between the noise-free image $s$ and the denoised image $u$. It is defined as.

$$\text{SSIM}(s,u) = \frac{1}{N_1N_2}\sum_{i=1}^{N_1N_2}\frac{\left(2\mu_{s_i}\mu_{u_i}+c_1\right)\left(2\sigma_{s_iu_i}+c_1\right)}{\left(\mu_{s_i}^2+\mu_{u_i}^2+c_1\right)\left(\sigma_{s_i}^2+\sigma_{u_i}^2+c_2\right)} \tag{26}$$

where $\mu_{s_i}$ and $\mu_{u_i}$ are the average of local image patch $s_i$ and $u_i$, respectively. $\sigma_{s_i}^2$, $\sigma_{u_i}^2$ are variances and $\sigma_{s_iu_i}$ is the covariance of $s_i$ and $u_i$. $c_1$ and $c_2$ are constants. Author Wang et al. of [23] suggests using the down sampling version SSIM. The corresponding Matlab code comes from https://ece.uwaterloo.ca/~z70wang/research/ssim/.

### 3.1 Role of spatial kernel $\phi_s$

The denoised image with $\omega_{cpr}=\phi_r$ is yielded from an estimate using the proposed method. Although the noise is removed, excessive smoothness affects the visual effect in a sense (see Fig. 3b). Then the denoised image with $\omega_{cpr}=\phi_s\phi_r$ avoids the disadvantage of excessive smoothness. The result in Fig. 3c is more natural than that in Fig. 3b (see the capsicum in the lower part of Figs. 3a, b and c). In addition, the PSNR and SSIM values of the denoised image in Fig. 3b are 28.45 dB and 0.8210, respectively; the PSNR and SSIM values of the denoised image in Fig. 3c are 28.67 dB and 0.8341, respectively. In a word, compared to the result without spatial kernel, the results using the proposed method with spatial kernel have different degrees of ascension in term of subjective and objective performance.

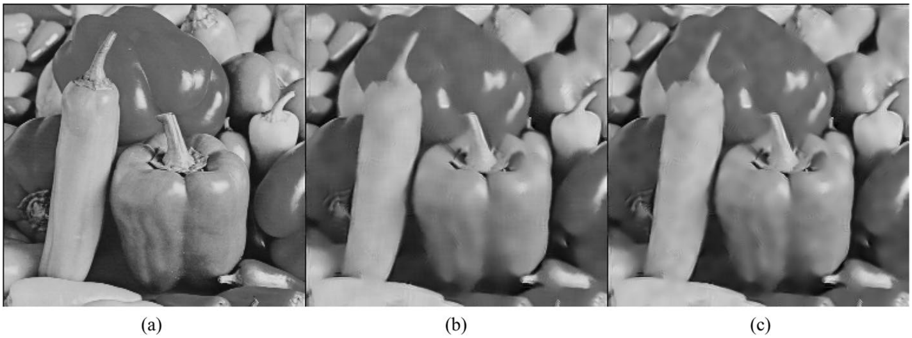

**Fig. 3** Comparison of denoised result without $\phi_s$and denoised result with $\phi_s$ using the proposed method for Peppers (256×256). **a** Original image; **b** denoised image without $\phi_s$; **c** denoised image with $\phi_s$

### 3.2 Comparison of the proposed method and the related NLM methods

Tables 2, 3 and 4 present the PSNR and SSIM results of NLM-Stein, NLM-LJS, NLM-Max, NLM-WFCW and the proposed method for$\sigma = 10$, 20, 30, respectively. One can find that the proposed method always outperforms other several methods in PSNR and SSIM in almost all cases. Except for Fingerprint and House at $\sigma = 10$, the proposed method is slightly inferior to the highest PSNR or SSIM among the obtained results. For Fingerprint, in PSNR performance, 31.09 dB with the proposed method is inferior to 31.19 dB with NLM-LJS. For House, in SSIM performance, 0.8963 with the proposed method is inferior to 0.8967 dB with NLM-LJS. However, in all other cases, the proposed method achieves the best results. Furthermore, the advantage of the proposed method becomes more obvious when the noise level increases.

Table 5 presents PSNR and SSIM results averaged over all the test images for every noise level $\sigma$ ($\sigma \in \{10, 20, 30\}$) and for all three noise levels for Proposed and several other NLM methods. It is noted that the center weight strategy of the proposed method comes from my previous work NLM-WFCW proposed in 2021. The comparison with NLM-WFCW is an ablation study. The comparison shows that the proposed method has weak improvement in noise level 10, while the improvement has become increasingly significant in noise levels 20 and 30. At the noise 30, compared to NLM-WFCW, the averaged gains are 0.40 dB in PSNR and 0.0128 in SSIM, respectively. For all three noise levels, compared to NLM-WFCW, the averaged gains are 0.24 dB in PSNR and 0.0085 in SSIM, respectively.

In a word, compared to other methods, the proposed method is more stable and excellent in PSNR and SSIM.

**Table 2** PSNR and SSIM values on Proposed and several other NLM methods for $\sigma = 10$

| | Index | NLM-Stein | NLM-LJS | NLM-Max | NLM-WFCW | Proposed |
|---|---|---|---|---|---|---|
| Cameraman | PSNR | 32.38 | 33.02 | 31.39 | 33.14 | 33.18 |
| | SSIM | 0.8996 | 0.9094 | 0.8948 | 0.9149 | 0.9165 |
| Lena | PSNR | 34.63 | 35.12 | 35.14 | 35.21 | 35.30 |
| | SSIM | 0.9531 | 0.9612 | 0.9594 | 0.9607 | 0.9627 |
| Montage | PSNR | 34.65 | 34.88 | 31.79 | 35.41 | 35.46 |
| | SSIM | 0.9446 | 0.9385 | 0.9495 | 0.9511 | 0.9519 |
| House | PSNR | 34.81 | 35.31 | 35.45 | 35.49 | 35.54 |
| | SSIM | 0.8812 | 0.8967 | 0.8938 | 0.8950 | 0.8963 |
| Peppers | PSNR | 33.02 | 33.49 | 33.39 | 33.91 | 33.98 |
| | SSIM | 0.9030 | 0.9067 | 0.9150 | 0.9186 | 0.9204 |
| Barbara | PSNR | 33.20 | 33.75 | 33.73 | 33.85 | 33.90 |
| | SSIM | 0.9646 | 0.9690 | 0.9686 | 0.9700 | 0.9712 |
| Boat | PSNR | 32.14 | 32.92 | 32.80 | 33.00 | 33.12 |
| | SSIM | 0.9342 | 0.9522 | 0.9466 | 0.9504 | 0.9540 |
| Hill | PSNR | 31.88 | 32.65 | 32.40 | 32.78 | 32.89 |
| | SSIM | 0.9129 | 0.9408 | 0.9310 | 0.9371 | 0.9417 |
| Couple | PSNR | 32.26 | 32.89 | 32.64 | 33.04 | 33.16 |
| | SSIM | 0.9351 | 0.9520 | 0.9468 | 0.9503 | 0.9541 |
| Fingerprint | PSNR | 30.19 | 31.19 | 31.01 | 31.08 | 31.09 |
| | SSIM | 0.9868 | 0.9871 | 0.9853 | 0.9874 | 0.9874 |

**Table 3** PSNR and SSIM values on Proposed and several other NLM methods for $\sigma = 20$

| | Index | NLM-Stein | NLM-LJS | NLM-Max | NLM-WFCW | Proposed |
|---|---|---|---|---|---|---|
| Cameraman | PSNR | 28.62 | 29.32 | 29.19 | 29.57 | 29.63 |
| | SSIM | 0.8136 | 0.8373 | 0.8272 | 0.8309 | 0.8399 |
| Lena | PSNR | 31.30 | 31.62 | 31.94 | 31.97 | 32.28 |
| | SSIM | 0.9071 | 0.9194 | 0.9198 | 0.9207 | 0.9269 |
| Montage | PSNR | 30.56 | 30.58 | 30.15 | 31.56 | 31.63 |
| | SSIM | 0.8980 | 0.8818 | 0.8999 | 0.8992 | 0.9052 |
| House | PSNR | 31.78 | 31.82 | 32.50 | 32.51 | 32.67 |
| | SSIM | 0.8383 | 0.8338 | 0.8432 | 0.8416 | 0.8468 |
| Peppers | PSNR | 29.59 | 29.78 | 30.37 | 30.55 | 30.73 |
| | SSIM | 0.8464 | 0.8415 | 0.8593 | 0.8596 | 0.8681 |
| Barbara | PSNR | 29.53 | 29.89 | 30.18 | 30.25 | 30.50 |
| | SSIM | 0.9119 | 0.9261 | 0.9259 | 0.9274 | 0.9335 |
| Boat | PSNR | 28.71 | 29.22 | 29.60 | 29.72 | 30.07 |
| | SSIM | 0.8489 | 0.8823 | 0.8802 | 0.8840 | 0.8997 |
| Hill | PSNR | 28.36 | 29.05 | 29.38 | 29.53 | 29.91 |
| | SSIM | 0.8130 | 0.8572 | 0.8569 | 0.8619 | 0.8799 |
| Couple | PSNR | 28.21 | 28.79 | 29.10 | 29.29 | 29.71 |
| | SSIM | 0.8450 | 0.8775 | 0.8782 | 0.8827 | 0.8996 |
| Fingerprint | PSNR | 26.33 | 26.87 | 27.43 | 27.49 | 27.73 |
| | SSIM | 0.9206 | 0.9468 | 0.9455 | 0.9480 | 0.9569 |

**Table 4** PSNR and SSIM values on Proposed and several other NLM methods for $\sigma = 30$

| | Index | NLM-Stein | NLM-LJS | NLM-Max | NLM-WFCW | Proposed |
|---|---|---|---|---|---|---|
| Cameraman | PSNR | 26.89 | 27.16 | 27.52 | 27.77 | 27.86 |
| | SSIM | 0.7723 | 0.7736 | 0.7694 | 0.7700 | 0.8013 |
| Lena | PSNR | 29.07 | 29.39 | 29.82 | 29.82 | 30.34 |
| | SSIM | 0.8628 | 0.8778 | 0.8796 | 0.8796 | 0.8876 |
| Montage | PSNR | 27.89 | 28.03 | 28.63 | 29.31 | 29.53 |
| | SSIM | 0.8535 | 0.8269 | 0.8444 | 0.8425 | 0.8821 |
| House | PSNR | 29.03 | 29.18 | 29.99 | 29.97 | 30.44 |
| | SSIM | 0.7907 | 0.7756 | 0.7862 | 0.7832 | 0.8157 |
| Peppers | PSNR | 27.00 | 27.33 | 28.21 | 28.28 | 28.69 |
| | SSIM | 0.7928 | 0.7803 | 0.8046 | 0.8037 | 0.8341 |
| Barbara | PSNR | 26.92 | 27.34 | 27.70 | 27.70 | 28.03 |
| | SSIM | 0.8569 | 0.8771 | 0.8746 | 0.8790 | 0.8884 |
| Boat | PSNR | 26.52 | 27.05 | 27.56 | 27.58 | 28.08 |
| | SSIM | 0.7836 | 0.8208 | 0.8240 | 0.8253 | 0.8423 |
| Hill | PSNR | 26.71 | 27.22 | 27.66 | 27.68 | 28.09 |
| | SSIM | 0.7458 | 0.7921 | 0.7986 | 0.8002 | 0.8174 |
| Couple | PSNR | 25.80 | 26.43 | 26.86 | 26.90 | 27.40 |
| | SSIM | 0.7588 | 0.8053 | 0.8094 | 0.8111 | 0.8320 |
| Fingerprint | PSNR | 23.67 | 24.31 | 25.05 | 25.06 | 25.61 |
| | SSIM | 0.8400 | 0.8901 | 0.8917 | 0.8927 | 0.9144 |

**Table 5** PSNR and SSIM results averaged over all the test images for every noise level $\sigma$ and for all three noise levels for Proposed and several other NLM methods

| $\sigma$ | Index | NLM-Stein | NLM-LJS | NLM-Max | NLM-WFCW | Proposed |
|---|---|---|---|---|---|---|
| 10 | PSNR | 32.92 | 33.52 | 32.79 | 33.69 | 33.76 |
| | SSIM | 0.0.9315 | 0.9414 | 0.9391 | 0.9436 | 0.9456 |
| 20 | PSNR | 29.30 | 29.69 | 29.98 | 30.24 | 30.49 |
| | SSIM | 0.8643 | 0.8804 | 0.8836 | 0.8856 | 0.8957 |
| 30 | PSNR | 26.95 | 27.34 | 27.90 | 28.01 | 28.41 |
| | SSIM | 0.8061 | 0.8220 | 0.8283 | 0.8387 | 0.8515 |
| Average | PSNR | 29.72 | 30.19 | 30.23 | 30.65 | 30.89 |
| | SSIM | 0.8673 | 0.8812 | 0.8830 | 0.8891 | 0.8976 |

Figures 4 and 5 present the visual comparisons of results of NLM-Stein, NLM-LJS, NLM-Max, NLM-WFCW and the proposed method.

In Fig. 4, noisy Barbara (512×512) image with $\sigma = 20$ is processed. Figures 4c, d, e, f and g are the denoised results with NLM-Stein, NLM-LJS, NLM-Max, NLM-WFCW and the proposed method, respectively. Figure 4h shows the bright dot block on table legs in the lower left corner with original Barbara, NLM-LJS, NLM-WFCW and the proposed method in turn from left to right. One notices that there is a rectangular area in the lower right corner of Fig. 4g for the proposed method. In Figs. 4c and e, the dot block in this area is invisible compared to that in Fig. 4g. This is because the smooth strength in the edges and details domain for the proposed method is reasonably reduced in the whole denoising process compared to NLM-Stein and NLM-Max. For NLM-LJS, NLM-WFCW and the proposed method, the area is well preserved. But, for the bright dot block on table legs in the lower left corner of Fig. 4a, the presentations with NLM-LJS, NLM-WFCW and the proposed method have some differences. The result with proposed method is more faithful to the original image. For example, for the original image, the mean and standard deviation of the bright dot block is 46.62 and 22.81, respectively. For the result Fig. 4d with NLM-LJS, the mean and standard deviation of the bright dot block is 46.08 and 10.83, respectively. For the result Fig. 4f with NLM-WFCW, the mean and standard deviation of the bright dot block is 44.78 and 10.97, respectively. For the result Fig. 4g with the proposed method, the mean and standard deviation of the bright dot block is 45.61 and 13.31, respectively. It can be see that the proposed method is closer to the original image features in term of standard deviation compared to NLM-LJS and NLM-WFCW. Although the mean with NLM-LJS is closer to that of original image, there are many obvious noise points in the result with NLM-LJS (see Fig. 4h). Visually, the bright dot block on table legs with the proposed method is brighter in comparison with that with NLM-WFCW and NLM-LJS. And noise is handled well with the proposed method (see Fig. 4h).

From Fig. 4, the proposed method is better in the preservation of image features. This merit is shown in Fig. 5. In which, noisy Peppers image with $\sigma = 10$ is processed. Figures. 5c, d, e, f and g are the denoised results with NLM-Stein, NLM-LJS, NLM-Max, NLM-WFCW and the proposed method, respectively. One can find that there is a rectangular area in the lower left corner of Fig. 5g for the proposed method. In Fig. 5e, the area emerges in the same position, but this area is over-smoothed compared to that in Fig. 5g. For NLM-Stein and NLM-LJS, the area is well preserved. But, there are the left noise in a lot of other areas in Figs. 5c and d because of the insufficient denoising. For NLM-WFCW and the proposed method, the good visual effect is achieved.

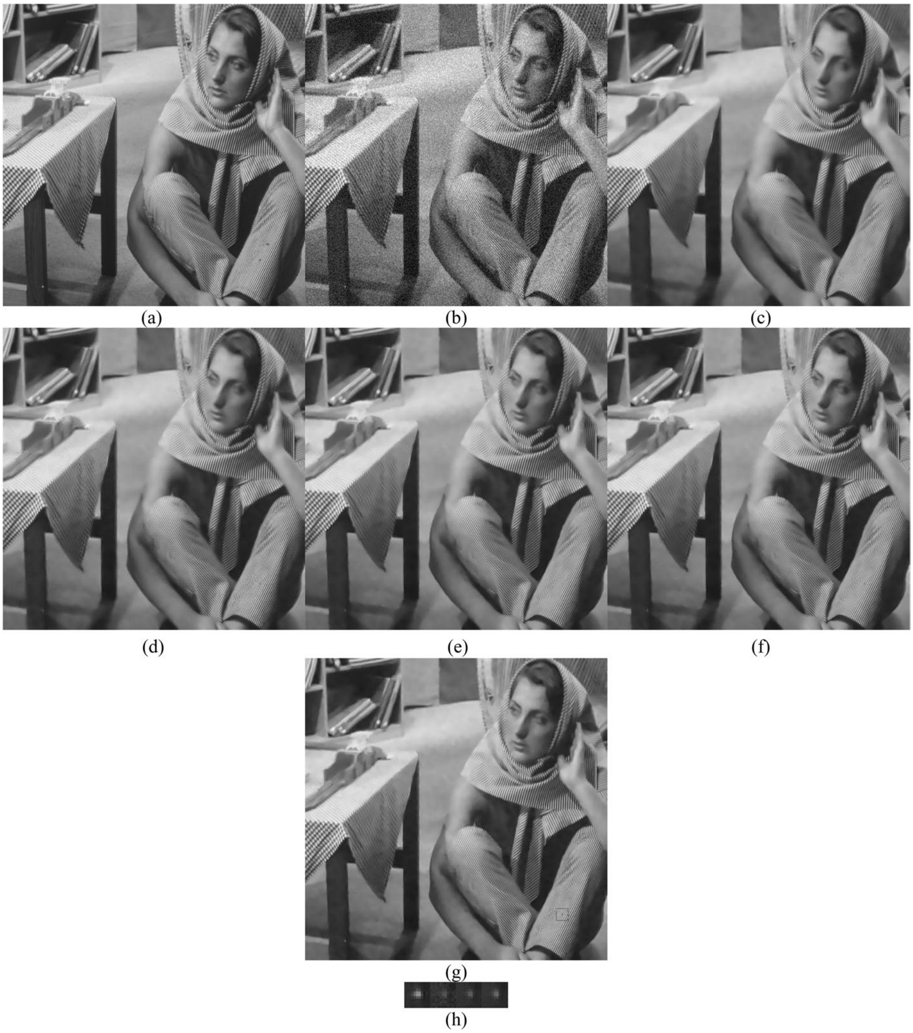
(a) (b) (c)
(d) (e) (f)
(g)
(h)

**Fig. 4** Comparison of the restoration results from the different methods. Zoom into file for a better view. **a** Original Barbara (512×512) image; **b** noisy image ($\sigma=20$); **c**-**g** shows restored Barbara images using NLM-Stein, NLM-LJS, NLM-Max, NLM-WFCW and the proposed method, respectively; **h** shows the bright dot block on table legs in the lower left corner with original Barbara, NLM-LJS, NLM-WFCW and the proposed method in turn from left to right

In a word, compared to other methods, the proposed method can better preserve textures, details and remove the noise. This is because the difference of pixels is used sufficiently in the proposed method.

NLM-Max is a classical method, and other NLM filters are produced by replacing the center weight (CW) of NLM-Max. Only for the proposed method, the spatial kernel need also be computed. Therefore, compared to the NLM-Max, for each pixel, the proposed method need only mainly a few addition and product operations in calculating

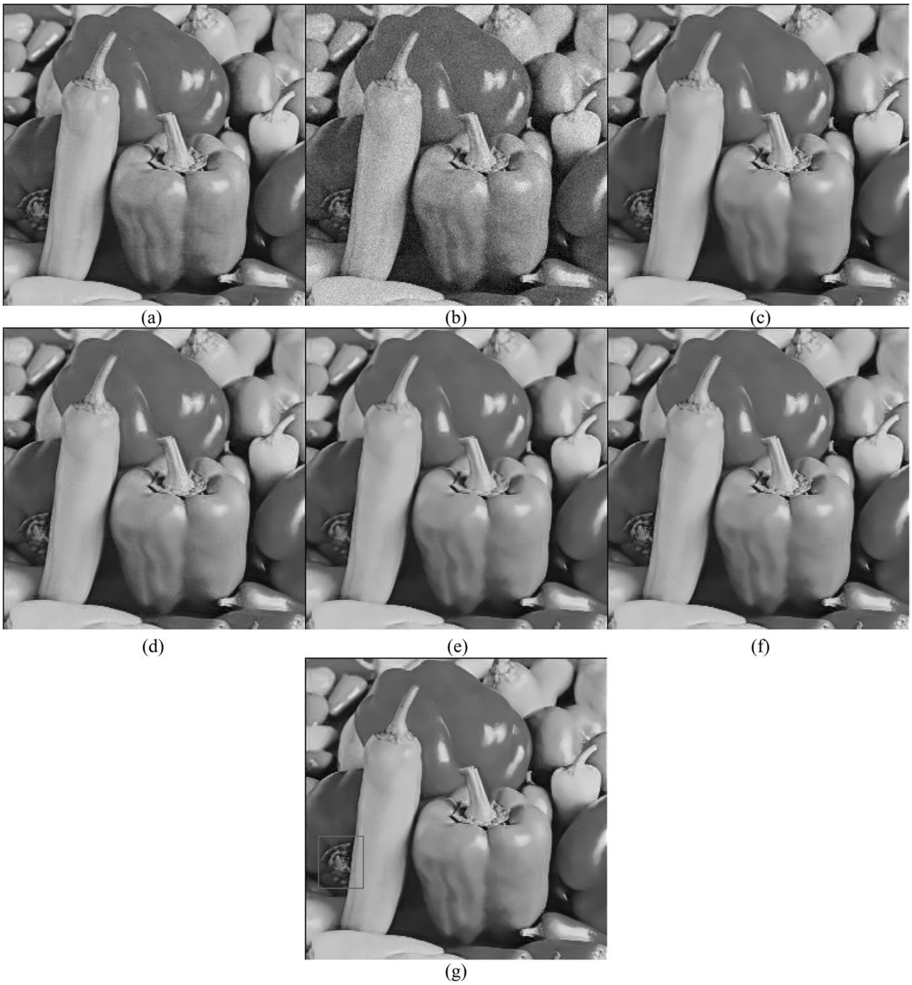

**Fig. 5** Comparison of the restoration results from the different methods. Zoom into file for a better view. **a** Original Peppers (256×256) image; **b** noisy image ($\sigma = 10$); **c**-**g** shows restored Peppers images using NLM-Stein, NLM-LJS, NLM-Max, NLM-WFCW and the proposed method, respectively

the variance of pixel in the proposed CW and spatial kernel (please see Eq. (19)), so the computational time increment for the proposed method is negligible. Here, for NLM-Max and the proposed method, the average running time is measured by code from [15]. For example, for the size 256×256 and 512×512 image, the running times with NLM-Max are 20.12 s and 80.33 s, respectively. The running times with the proposed method are 20.13 s and 80.34 s, respectively. It can be said that the proposed method has achieved good performance with little increase in computing time.

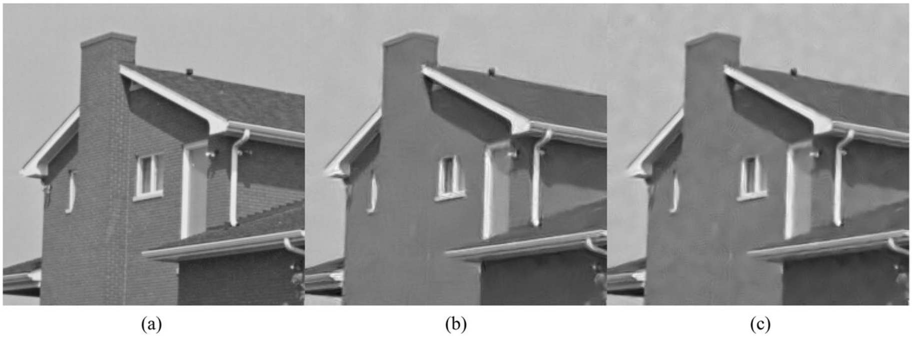

**Fig. 6** Comparison of the restoration results from NLM-Fov and Proposed for noise level $\sigma = 20$. Zoom into file for a better view. **a** Original House (256 × 256) image; **b**-**c** shows restored House images using NLM-Fov and the proposed method, respectively

### 3.3 Comparison of the proposed method and two state-of-the-art NLM methods

To further verify the effectiveness of the proposed method, two state-of-the-art NLM filters are used to compare. Here, the compared code presented by the corresponding authors is written in a similar form to that of [25]. For the sake of fair comparison, the running time is measured by modified code from [25] for the proposed method.

Figures 6, 7 and Tables 6 and 7 present the comparative results between NLM-Fov and Proposed. Visually, the result with NLM-Fov looks smoother in the case of shrinking and appears many minor false details in case of magnification (see Figs. 6 and 7). Although the proposed method show the block effects, it better preserves the contrast of structure, has less artifacts around edges and looks more natural (see the capsicum in the lower part of Figs. 7a, b and c). In a word, two methods have their own advantages. These have been embodied in PSNR and SSIM in Tables 6 and 7. The performance of two methods is comparative. Although the foveated NLM filter has shown the state-of-the-art performance, the current research demonstrates that the proposed window based NLM filter by specific design also can compete with the foveated NLM filter. Furthermore, the proposed

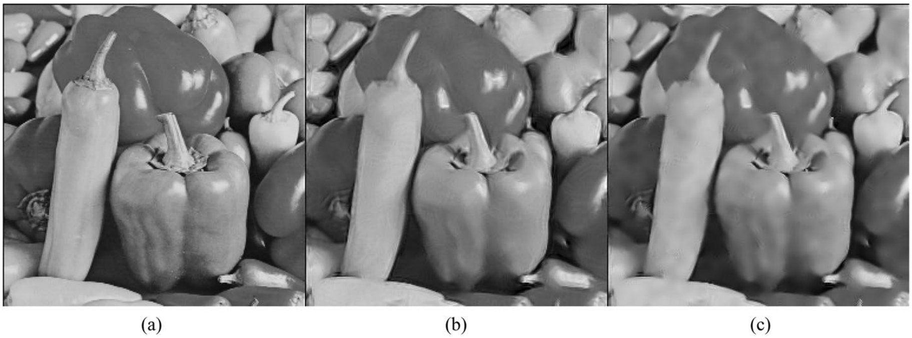

**Fig. 7** Comparison of the restoration results from NLM-Fov and Proposed for noise level $\sigma = 30$. Zoom into file for a better view. **a** Original Peppers (256 × 256) image; **b**-**c** shows restored Peppers images using NLM-Fov and the proposed method, respectively

**Table 6** PSNR and SSIM values on Proposed and NLM-Fov for $\sigma = 10, 20, 30$

| | Index | NLM-Fov | Proposed | NLM-Fov | Proposed | NLM-Fov | Proposed |
|---|---|---|---|---|---|---|---|
| $\sigma$ | | 10 | | 20 | | 30 | |
| Cameraman | PSNR | 31.70 | 33.18 | 29.26 | 29.63 | 27.81 | 27.86 |
| | SSIM | 0.9064 | 0.9165 | 0.8396 | 0.8399 | 0.7931 | 0.8013 |
| Lena | PSNR | 35.06 | 35.30 | 32.43 | 32.28 | 30.68 | 30.34 |
| | SSIM | 0.9633 | 0.9627 | 0.9315 | 0.9269 | 0.9006 | 0.8876 |
| Montage | PSNR | 33.62 | 35.46 | 30.59 | 31.63 | 29.00 | 29.53 |
| | SSIM | 0.9496 | 0.9519 | 0.9093 | 0.9052 | 0.8698 | 0.8821 |
| House | PSNR | 35.41 | 35.54 | 32.90 | 32.67 | 31.13 | 30.44 |
| | SSIM | 0.8977 | 0.8963 | 0.8526 | 0.8468 | 0.8176 | 0.8157 |
| Peppers | PSNR | 33.38 | 33.98 | 30.51 | 30.73 | 28.68 | 28.69 |
| | SSIM | 0.9178 | 0.9204 | 0.8667 | 0.8681 | 0.8225 | 0.8341 |
| Barbara | PSNR | 33.60 | 33.90 | 30.31 | 30.50 | 27.97 | 28.03 |
| | SSIM | 0.9714 | 0.9712 | 0.9358 | 0.9335 | 0.8935 | 0.8884 |
| Boat | PSNR | 32.85 | 33.12 | 30.12 | 30.07 | 28.35 | 28.08 |
| | SSIM | 0.9554 | 0.9540 | 0.9018 | 0.8997 | 0.8521 | 0.8423 |
| Hill | PSNR | 32.82 | 32.89 | 30.07 | 29.91 | 28.44 | 28.09 |
| | SSIM | 0.9455 | 0.9417 | 0.8816 | 0.8799 | 0.8267 | 0.8174 |
| Couple | PSNR | 32.77 | 33.16 | 29.93 | 29.71 | 27.96 | 27.40 |
| | SSIM | 0.9566 | 0.9541 | 0.9067 | 0.8996 | 0.8529 | 0.8320 |
| Fingerprint | PSNR | 30.37 | 31.09 | 27.52 | 27.73 | 25.71 | 25.61 |
| | SSIM | 0.9851 | 0.9874 | 0.9609 | 0.9569 | 0.9249 | 0.9144 |

method can hardly increase the calculation time compared with NLM-Max. But NLM-Fov increases the calculation time by over 50% compared with NLM-Max (For fair comparison, the implementation by NLM-Max comes from code of NLM-Fov). This is because that the proposed method only uses a Gaussian filter in the computation of weight but NLM-Fov employs more filters in essence.

Figures 8, 9 and Tables 8 and 9 present the comparative results between NLM-SNN and Proposed. Figure 8 with the proposed method produces more obvious artifacts around the edges (see "Hello World" in Fig. 8c). But, Fig. 9c with proposed method better preserves the features and contrasts in camera compared to Fig. 9b with NLM-SNN. Likewise, from Tables 8 and 9, these have been embodied in PSNR and SSIM. In general, the performance of the two methods is comparable. They have their own characteristics and strength. Furthermore, the proposed method can produce the result with less time. For NLM-SNN,

**Table 7** PSNR and SSIM results averaged over all the test images for every noise level $\sigma$ and for all three noise levels for Proposed and NLM-Fov

| $\sigma$ | Index | NLM-Fov | Proposed |
|---|---|---|---|
| 10 | PSNR | 33.16 | 33.76 |
| | SSIM | 0.9449 | 0.9456 |
| 20 | PSNR | 30.36 | 30.49 |
| | SSIM | 0.9005 | 0.8957 |
| 30 | PSNR | 28.57 | 28.41 |
| | SSIM | 0.8554 | 0.8515 |
| Average | PSNR | 30.70 | 30.89 |
| | SSIM | 0.9002 | 0.8976 |

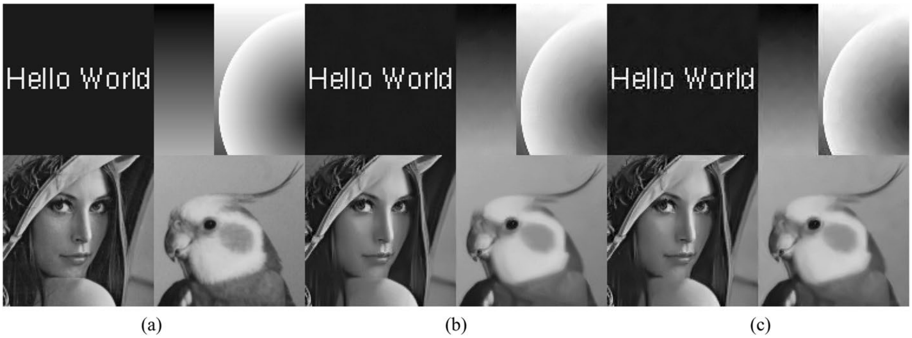

**Fig. 8** Comparison of the restoration results from NLM-SNN and Proposed for noise level $\sigma = 10$. Zoom into file for a better view. **a** Original Montage (256×256) image; **b**-**c** shows restored Montage images using NLM-SNN and the proposed method, respectively

selection of patch and aggregation of multiple estimates costs more times. For example, for the size 256×256 and 512×512 image, the average running times with the proposed method are about 0.70 s and 2.68 s, respectively. But, for the size 256×256 and 512×512 image, the average running times with NLM-SNN are 9.02 s and 62.05 s, respectively. Obviously, the proposed method saves plenty of time.

By comparison, the proposed method is superior to the several excellent NLM filters such as NLM-Stein, NLM-LJS, NLM-Max and NLM-WFCW in subjective and objective performance. At the same time, the proposed method is also a state-of-the-art NLM filter by the comparative results with NLM-Fov and NLM-SNN.

## 4 Conclusion

By exploring the spatial kernel to remedy the statistical nearest neighbor distance measurement (SNNDM) in the weight computation in NLM filter, in this paper one proposes a bilateral thresholding to calculate the weight of each pixel in neighbor at the reference

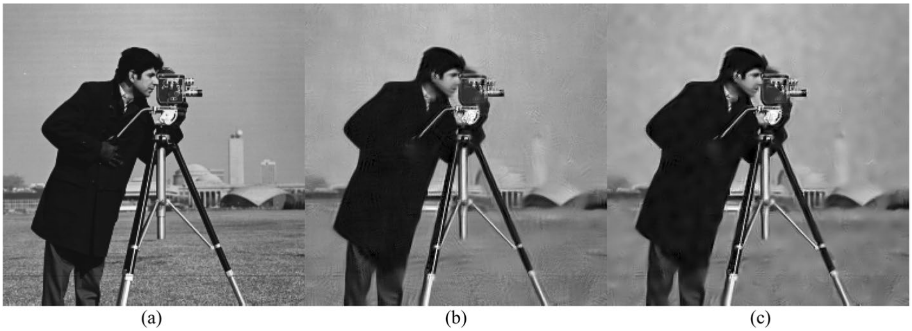

**Fig. 9** Comparison of the restoration results from NLM-SNN and Proposed for noise level $\sigma = 30$. Zoom into file for a better view. **a** Original Cameraman (256×256) image; **b**-**c** shows restored Cameraman images using NLM-SNN and the proposed method, respectively

**Table 8** PSNR and SSIM values on Proposed and NLM-SNN for $\sigma = 10, 20, 30$

| | Index | NLM-SNN | Proposed | NLM-SNN | Proposed | NLM-SNN | Proposed |
|---|---|---|---|---|---|---|---|
| $\sigma$ | | 10 | | 20 | | 30 | |
| Cameraman | PSNR | 33.59 | 33.18 | 29.73 | 29.63 | 27.68 | 27.86 |
| | SSIM | 0.9197 | 0.9165 | 0.8399 | 0.8399 | 0.7723 | 0.8013 |
| Lena | PSNR | 35.18 | 35.30 | 32.00 | 32.28 | 30.01 | 30.34 |
| | SSIM | 0.9615 | 0.9627 | 0.9203 | 0.9269 | 0.8797 | 0.8876 |
| Montage | PSNR | 36.40 | 35.46 | 32.38 | 31.63 | 29.69 | 29.53 |
| | SSIM | 0.9572 | 0.9519 | 0.9069 | 0.9052 | 0.8492 | 0.8821 |
| House | PSNR | 35.78 | 35.54 | 32.63 | 32.67 | 30.48 | 30.44 |
| | SSIM | 0.9025 | 0.8963 | 0.8360 | 0.8468 | 0.7808 | 0.8157 |
| Peppers | PSNR | 34.13 | 33.98 | 30.45 | 30.73 | 28.26 | 28.69 |
| | SSIM | 0.9184 | 0.9204 | 0.8561 | 0.8681 | 0.7968 | 0.8341 |
| Barbara | PSNR | 33.98 | 33.90 | 30.73 | 30.50 | 28.51 | 28.03 |
| | SSIM | 0.9704 | 0.9712 | 0.9361 | 0.9335 | 0.8934 | 0.8884 |
| Boat | PSNR | 33.05 | 33.12 | 29.77 | 30.07 | 27.77 | 28.08 |
| | SSIM | 0.9536 | 0.9540 | 0.8903 | 0.8997 | 0.8287 | 0.8423 |
| Hill | PSNR | 32.83 | 32.89 | 29.65 | 29.91 | 27.86 | 28.09 |
| | SSIM | 0.9412 | 0.9417 | 0.8666 | 0.8799 | 0.8000 | 0.8174 |
| Couple | PSNR | 33.29 | 33.16 | 29.55 | 29.71 | 27.39 | 27.40 |
| | SSIM | 0.9561 | 0.9541 | 0.8914 | 0.8996 | 0.8239 | 0.8320 |
| Fingerprint | PSNR | 30.79 | 31.09 | 27.29 | 27.73 | 25.30 | 25.61 |
| | SSIM | 0.9876 | 0.9874 | 0.9523 | 0.9569 | 0.9127 | 0.9144 |

noisy pixel. In order to make full and proper use of the differences of different pixels, the center weight and other weights are also treated differently. Some simulations are taken on several grayscale natural images to investigate the performance of the proposed method. The results prove the proper and simple weight function design can bring superiority with negligible differences in calculation time to other related NLM filter approaches such as NLM-Stein, NLM-LJS, NLM-Max and NLM-WFCW in PSNR, SSIM and visual performance. Furthermore, the proposed method is on par with the-state-of-the-art NLM methods such as NLM-Fov and NLM-SNN in PSNR, SSIM and visual comprehensive performance, but the proposed method can save plenty of running time. In addition, it is noted that the proposed pixel wise NLM method originates from the patch wise SNNDM and other analysis. In the future, one will seek more appropriate

**Table 9** PSNR and SSIM results averaged over all the test images for every noise level $\sigma$ and for all three noise levels for Proposed and NLM-SNN

| $\sigma$ | Index | NLM-SNN | Proposed |
|---|---|---|---|
| 10 | PSNR | 33.91 | 33.76 |
| | SSIM | 0.9468 | 0.9456 |
| 20 | PSNR | 30.42 | 30.49 |
| | SSIM | 0.8896 | 0.8957 |
| 30 | PSNR | 28.29 | 28.41 |
| | SSIM | 0.8337 | 0.8515 |
| Average | PSNR | 30.88 | 30.89 |
| | SSIM | 0.8900 | 0.8976 |

distance measurement ways to compute the weight of neighbor pixel to improve patch wise NLM 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).

**Data availability** Data will be made available on request.

## Declarations

**Conflict of interest** The author declares that he has no conflict of interest.

## References

1. Buades A, Coll B, Morel JM (2005) A non-local algorithm for image denoising. In: 2005 IEEE computer society conference on computer vision and pattern recognition. IEEE, San Diego, CA, USA, pp 60–65
2. Dabov K, Foi A, Katkovnik V (2007) Image denoising by sparse 3-D transformdomain collaborative filtering. IEEE Trans Image Process 16(8):2080–2095
3. De la Rosa JI, Villa-Hernández J, Cortez J, Gamboa H, Arceo JG, González E (2018) On the comparison of different kernel functionals and neighborhood geometry for nonlocal means filtering. Multimed Tools Appl 77(1):1205–1235
4. Fedorov V, Ballester C (2017) Affine non-local means image denoising. IEEE Trans Image Process 26(5):2137–2148
5. Foi A, Boracchi G (2016) Foveated nonlocal self-similarity. Int J Comput Vis 120(1):78–110
6. Frosio I, Kautz J (2019) Statistical nearest neighbors for image denoising. IEEE Trans Image Process 28(2):723–738
7. Ghosh S, Chaudhury KN (2016) Fast separable nonlocal means. J Electron Imaging 25(2):023026
8. Gu S, Xie Q, Meng D, Zuo W, Feng X, Zhang L (2017) Weighted nuclear norm minimization and its applications to low level vision. Int J Comput Vis 121(2):183–208
9. Heidelberg B, Goossens H, Luong A, Pizurica W (2008) An improved non-local denoising algorithm. In: 2008 International Workshop on Local and Non-Local Approximation in Image Processing. Lausanne, Switzerland, pp 143–156
10. Hue NM, Thanh DN, Hien NN, Prasath VS (2019) Image denoising with overlapping group sparsity and second order total variation regularization. In: 2019 6th IEEE, NAFOSTED Conference on Information and Computer Science. IEEE, Hanoi, Vietnam, pp 370–374
11. Kumain SC, Singh M, Singh N, Kumar K (2018) An efficient Gaussian noise reduction technique for noisy images using optimized filter approach. In: 2018 First IEEE International Conference on Secure Cyber Computing and Communication. IEEE, Jalandhar, India, pp 243–248
12. Kumar S, Kumar S, Shafi M, Chaube MK (2022) A novel multimodal framework for automatic recognition of individual cattle based on hybrid features using sparse stacked denoising autoencoder and group sparse representation techniques. Multimed Tools Appl 81(21):31075–31106
13. Lefkimmiatis S (2017) Non-local color image denoising with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, Honolulu, HI, USA, pp. 3587–3596
14. Li X, Zhou Y, Zhang J, Wang L (2019) Multipatch unbiased distance non-local adaptive means with wavelet shrinkage. IEEE Trans Image Process 29:157–169
15. Manjon-Herrera JV (2008) Non-local means filter, Matlab code. Matlab Central File Exchange http://www.mathworks.com/matlabcentral/fileexchange/13176-non-local-means-filter
16. 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
17. Qu Y, Swamy MNS, Luo J, Li B (2022) Single image denoising via multi-scale weighted group sparse coding. Signal Process 200:108650
18. Salmon J (2009) On two parameters for denoising with non-local means. IEEE Signal Process Lett 17(3):269–272

19. Thanh DN, Prasath VS, Hieu LM, Dvoenko S (2020) An adaptive method for image restoration based on high-order total variation and inverse gradient. Signal Image Vid P 14(6):1189–1197
20. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. In: 6th IEEE International Conference on Computer Vision. IEEE, Bombay, India, pp 839–846
21. Van De Ville D, Kocher M (2009) SURE-based non-local means. IEEE Signal Process Lett 16(11):973–976
22. Vignesh R, Oh BT, Kuo CCJ (2009) Fast non-local means (NLM) computation with probabilistic early termination. IEEE Signal Process Lett 17(3):277–280
23. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error measurement to structural similarity. IEEE Trans Image Process 13(1):600–612
24. Wang Z, Ng MK, Zhuang L, Gao L, Zhang B (2022) Nonlocal self-similarity-based hyperspectral remote sensing image denoising with 3D convolutional neural network. IEEE T Geosci Remote 15:9293–9310
25. Wu Y (2022) James stein type center pixel weights for non-local means, Matlab code. Matlab Central File Exchange https://www.mathworks.com/matlabcentral/fileexchange/40162-james-stein-type-center-pixel-weights-for-non-local-means
26. Wu Y, Tracey B, Natarajan P, Noonan JP (2013) James–Stein type center pixel weights for non-local means image denoising. IEEE Signal Process Lett 20(4):411–414
27. Zhang X (2021) Center pixel weight based on wiener filter for non-local means image denoising. Optik 244:167557
28. Zhang X (2021) Image denoising using multidirectional gradient domain. Multimed Tools Appl 80(19):29745–29763
29. Zhang X (2022) Two-step non-local means method for image denoising. Multidim Syst Sign P 33(2):341–366
30. Zhang X, Feng X (2015) Image denoising using local adaptive layered wiener filter in the gradient domain. Multimed Tools Appl 74(23):10495–10514
31. Zhang Y, Wu J, Ding M, Zhang X (2019) Nonlocal means image denoising with minimum MSE-based decay parameter adaptation. IEEE Access 7:130246–130261

**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.