Hindawi
Scientific Programming
Volume 2023, Article ID 8855652, 19 pages
https://doi.org/10.1155/2023/8855652

Research Article

Dual-Core Adaptive NLM Image Denoising Algorithm Based on Variable-Size Window and Neighborhood Multifeatures

Jing Mao,[1] Lianming Sun,[2] Jie Chen,[3] and Shunyuan Yu[3]

[1] *Graduate School of Environmental Engineering, The University of Kitakyushu, Kitakyushu 808-0135, Japan*
[2] *Department of Information Systems Engineering, The University of Kitakyushu, Kitakyushu 808-0135, Japan*
[3] *School of Electronic and Information Engineering, Ankang University, Ankang 725000, China*

Correspondence should be addressed to Jing Mao; yuehediao@163.com

Received 19 April 2023; Revised 30 July 2023; Accepted 12 August 2023; Published 9 September 2023

Academic Editor: Sadiq Hussain

To solve the problem that the similarity calculation between neighbors was easily disturbed by noise in the traditional nonlocal mean (NLM) denoising algorithm, a dual-core NLM denoising algorithm based on neighborhood multifeatures and variable-size search window was proposed. The algorithm first proposed to use the eigenvalues of the structure tensor to classify the region where the target pixel points were located and used different sizes of the search window to search for similar neighborhoods for target pixel points in different categories of the region, thus effectively avoiding the problem of oversmoothing or inadequate denoising of the image caused by the use of the global size. Then, the gradient features between image blocks were defined and combined with grayscale features and spatial features to measure the similarity of neighborhood blocks, which solved the problem of noise interfering with the search of similar blocks. Then, an adaptive algorithm with Gaussian–Sinusoidal dual kernel function and quantitative estimation of the optimal values of the filtering parameters was designed to calculate the neighborhood similarity weights to improve the accuracy of image denoising. Finally, the similarity weights were used to weight and average the search neighborhood of the target pixel points to achieve the denoising of the target pixel points. To test the effectiveness of the algorithm, denoising tests were performed using multiple standard grayscale images with different levels of Gaussian white noise added and compared with several advanced denoising algorithms. The experimental results showed that the algorithm was effective. The algorithm improved the image peak signal-to-noise ratio by more than 56.54% on average when Gaussian white noise was removed, and the structural similarity reached more than 0.701 on average. Compared with the traditional NLM algorithm and other improved algorithms, the algorithm proposed in this paper had strong denoising ability, better protection of edges and texture details, and the quality of the image was greatly improved, which had a good application prospect.

1. Introduction

As a kind of information carrier, the digital image was an essential auxiliary technology in daily life, teaching, and research. Due to the influence of objective factors such as equipment and working environment, the digital image inevitably mixed with noise, which was not conducive to the application of the digital image. The noise in most images comes mainly from Gaussian noise and impulse noise. Gaussian noise was caused by sensor noise and electronic circuit noise under low illumination or high-temperature conditions. Gaussian noise was mainly characterized by a relatively high density and a wide range of intensity fluctuations. Impulse noise generated by the signal noise intensity was a collective term for discrete types of noise that appeared in communication, also called pretzel noise. When the real signal of the image itself was mixed with the noise signal, it would cause a degradation of the image quality, which would seriously affect the subsequent processing of the image and the application of the image. Therefore, removing noise from digital images was an important way to improve image quality.

Commonly available image denoising algorithms included spatial domain denoising algorithms, such as mean filtering algorithm [1], median filtering algorithm [2], Gaussian filtering algorithm [3], bilateral filtering algorithm [4], etc.; and transform domain algorithms, such as K–L transform algorithm [5], partial differential equation algorithm [6], wavelet transform algorithm [7], independent component analysis-based algorithm [8], various joint denoising methods and improved algorithms [9], etc. The abovementioned algorithms targeted individual pixels in an image as the research target, which was a denoising method in local mode and suffered from the problem of local feature leakage, such as image structure texture, which was difficult to meet the current digital image processing requirements.

To further improve the image denoising effect, Buades et al. [10] proposed the nonlocal mean denoising algorithm (nonlocal means, NLM); however, the algorithm was highly complex, computationally intensive, and prone to noise interference, so the denoising effect was not very satisfactory. In response to these problems, various improvement algorithms have been proposed; for example, Vignesh et al. [11] used Gaussian and sinusoidal functions instead of the exponential kernel function in the traditional NLM function. Due to the limitations of the algorithm principle, details such as the edge texture of the image were damaged. Yuzheng et al. [12] used multiangle gradient features to solve the problem of noise interference in similar block searches; however, the appropriate parameters need to be tested several times. Haiqing and Hongwei [13] used an improved hybrid robust weight function to calculate the image block similarity weights and combined it with a two-stage denoising framework to achieve the denoising purpose. The predicted image parameters were difficult to find accurately, making the denoising results need to be improved. Han et al. [14] combined neural networks with nonlocal data of images as the input of the network and used the trained network to achieve noise removal. Due to the convergence speed and accuracy of the network, its effectiveness could be further improved. Zhuhua et al. [15] proposed the detail-preserving iterative NLM method, which achieved better results; however, due to the variability of regions, the above method could be improved for removing noise at retail locations such as edges/textures. Shervan et al. [16] proposed an approach for noise reduction in texture images, which was based on real word spelling correction theory in natural language processing. The proposed approach first generated the most similar pixels to noisy desired pixels in terms of textural features using local binary patterns. Then, the best one of the candidates was selected based on two-gram algorithm. Asem [17] proposed an image noise removal method based on collaborative filtering and singular value decomposition. It used the singular value shrinkage to guarantee the elimination of highly noisy pixels and used the random matrix to practically choose the level of the singular value threshold. Khmag [18] used a generative adversarial network model and a semisoft thresholding method to achieve image noise removal [18], but it was susceptible to interference from external factors when carrying out noise removal due to the inability to produce diverse results when considering independent samples alone. Khmag [19] used a patch-based estimation technique to estimate noise level and applied it to the proposed blind image denoising algorithm. Since the principal component analysis was used to estimate the noise level of the selected patch, the method may underestimate the noise level, which can affect the denoising effect. In summary, the NLM algorithm still had certain limitations, particularly in the complex structure of digital image processing; there was a large room for improvement.

This paper improved the traditional NLM algorithm idea to increase the denoising ability of the algorithm and the ability of image texture and detail preservation. The main contributions were as follows:

(1) A method for classifying image regions based on the eigenvalues of the image structure tensor was proposed to select different sizes of search windows and adaptive smoothing filtering parameters for denoising images in different categories of regions, thus reducing the discrepancy between the denoising results and the original image.

(2) Based on other literature studies, this paper enriched and improved the definition of the gradient feature and combined it with the grayscale feature and spatial feature to measure the similarity of neighborhood fast, which solved the problem that noise interferes with the search of similar blocks.

(3) A Gaussian-sinusoidal dual kernel function was proposed to calculate the similarity weights, and an adaptive method was designed to quantitatively estimate the optimal values of the filtering parameters, which improved the accuracy of image denoising.

Simulation results showed that the method in this paper was superior to the traditional NLM method and other improved NLM methods in terms of enhancing image quality and improving denoising effect.

2. Overview of NLM Image Denoising Algorithms

The pixel gray value change was a direct reflection of the image color change and a quantitative display of the image texture detail transformation, which was the most direct way to measure the similarity [20], so the traditional NLM method used the neighborhood grayscale feature to measure the neighborhood similarity [21], whose formula was shown as follows:

$$d_{\text{gray}}(x, y) = \|N(x) - N(y)\|_2^2. \tag{1}$$

Here $\boldsymbol{N(x)}, \boldsymbol{N(y)}$ represent the pixel neighborhood block centered at pixel point x, pixel point y, respectively, and d_{gray} represents the grayscale similarity value of two-pixel blocks.

The denoised estimate $\widehat{I}(x)$ of an image pixel point $I(x)$ affected by noise could be expressed as follows:

$$\widehat{I}(x) = \sum_{y \in S_x} W(x, y) * I(y). \tag{2}$$

Here $I(y)$ represents the gray value of point y in the noisy image, $\boldsymbol{S_x}$ represents the search neighborhood centered on pixel x, and $W(x, y)$ represents the similarity weight between the two neighborhoods centered on pixel x and pixel y, respectively, which was expressed as follows:

$$W(x,y) = \frac{1}{\sum_{y \in S_x} W(x,y)} \exp^{-\frac{d_{\text{gray}}}{h^2}}. \tag{3}$$

In Equation (3), h represents the smoothing parameter that determines the degree of filtering of the algorithm.

The traditional NLM algorithm used an exponential function to calculate the weight value of a similar neighborhood. The exponential function had higher weights in the region of high similarity, but its weights decay too fast with the increase of Euclidean distance, which reduced the accuracy of weight assignment and led to the blurring of denoised images. To solve this problem, literature [11] used the Gaussian function, sine function, and other kernel functions to replace the exponential function in the traditional NLM algorithm. The kernel function formula was as follows:

$$W_g(x,y) = \exp\left(-\frac{d_{\text{gray}}^2}{h^2}\right), \tag{4}$$

$$W_d(x,y) = \begin{cases} \dfrac{\sin\left(\frac{\pi d_{\text{gray}}}{h}\right)}{\pi d_{\text{gray}} h} & d_{\text{gray}} < h \\ 0 & \text{others} \end{cases}. \tag{5}$$

Here h was the smoothing parameter; d_{gray} represents the grayscale Euclidean distance between pixel neighborhoods as shown in Equation (1). $W_g(x,y)$ represents the similarity weight between the two neighborhoods centered on pixel point x and pixel point y computed using the Gaussian kernel function, and $W_d(x,y)$ represents the similarity weight between the two neighborhoods computed using the sinusoidal kernel function.

The Gaussian kernel decreased rapidly with increasing distance, and when the Euclidean distance of the neighborhood was larger, the Euclidean distance had less influence on the similarity weight, and the similarity was close to zero, whereas the sinusoidal kernel had a slow decreasing trend within a certain Euclidean distance, and when the Euclidean distance was larger than a certain threshold its similarity weight was zero, and the neighborhoods with lower similarity were filtered out.

The improved NLM algorithm using several different kernel functions proposed in literature [11] improved the problems of the traditional NLM algorithm. However, the experimental results showed that the algorithm obtained a better filtering effect than the traditional NLM algorithm when the noise intensity was weak. But when the image noise was strong, the improved NLM algorithm with Gaussian kernel function would reduce the denoising performance due to insufficient weighting, while the improved NLM algorithm with sinusoidal kernel function would easily cause blurring of the image due to overweighting. Therefore, the above algorithms need to be further improved.

3. Improved Nonlocal Mean Denoising Algorithm

3.1. Selection of Variable Size Windows. It was found that using a small search window in the smoothed region would result in a larger difference between the pixel variance of the denoised result and the original image. Based on the above understanding, different search window sizes should be used for different regions to reduce the difference between the filtered results and the original image.

The structure tensor of an image can effectively reflect the current distribution of details, and for each pixel point (x,y) of the region, the structure tensor matrix was defined as follows:

$$\begin{aligned} \boldsymbol{T}_\sigma(x,y) &= \begin{bmatrix} t_{11} & t_{12} \\ t_{12} & t_{22} \end{bmatrix} \\ &= \begin{bmatrix} G_\sigma * (g_x(x,y))^2 & G_\sigma * g_x(x,y) * g_y(x,y) \\ G_\sigma * g_y(x,y) * g_x(x,y) & G_\sigma * \left(g_y(x,y)\right)^2 \end{bmatrix}. \end{aligned} \tag{6}$$

In Equation (6), g_x, g_y represent the gradient in the X and Y directions, respectively, and G_σ represents the Gaussian kernel in terms of standard deviation, the eigenvalues of the structure tensor can be expressed as follows:

$$\lambda_1(x,y) = \frac{1}{2}\left(t_{11} + t_{22} + \sqrt{(t_{11} - t_{22})^2 + 4t_{12}^2}\right), \tag{7}$$

$$\lambda_2(x,y) = \frac{1}{2}\left(t_{11} + t_{22} - \sqrt{(t_{11} - t_{22})^2 + 4t_{12}^2}\right). \tag{8}$$

Pixels in smooth areas had small differences in eigenvalues; pixels in edge/texture areas had larger differences in eigenvalues. Thus, image region classification can be achieved by detecting differences in the eigenvalues of each pixel. Define a region feature value λ and let $\lambda(x,y) = |\lambda_1(x,y) - \lambda_2(x,y)|$, then the whole image region can be divided into n categories (e.g., detail region, transition region, smooth region, etc.), and the classification judgment expression was as follows:

$$I(x,y) \in \begin{cases} A_1, & \lambda(x,y) \le \lambda_{\min} + \dfrac{(\lambda_{\max} - \lambda_{\min})}{n} \\ A_2, & \lambda(x,y) \le \lambda_{\min} + \dfrac{2 * (\lambda_{\max} - \lambda_{\min})}{n} \\ \vdots & \vdots \\ A_n, & \lambda(x,y) \le \lambda_{\min} + \dfrac{n * (\lambda_{\max} - \lambda_{\min})}{n} \end{cases}. \tag{9}$$

Here $\lambda_{\max}$, $\lambda_{\min}$ represent the maximum and minimum eigenvalues of the entire image structure tensor. $A_k, k = 1, 2, \cdots, n$ represent the range of regions for different categories, with smaller k indicating smoother regions and vice versa, and n represents the number of categories. Based on the

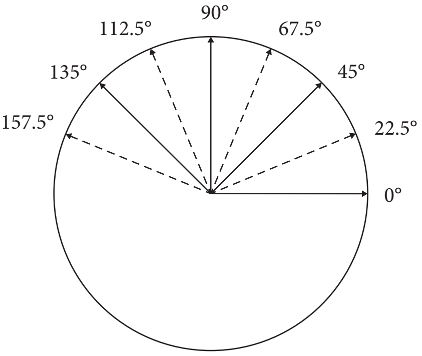


Figure 1: Schematic diagram of the different gradient orientations.

region classification results, different sizes of search windows were selected in different regions. Theoretically, the number of classified regions should be as large as possible so that the image partition can be divided more carefully. In practice, when n was larger than 4, the variation of the denoising effect tended to be unchanged. Therefore, in this paper, the number of partitions $n = 3$ was chosen in the experiment.

3.2. Neighborhood Characteristics. Literature [12] defined gradient features for 0° direction, 45° direction, 90° direction, and 135° direction, but the gradient features for other directions were not included in the similarity measure; therefore, this paper added gradient features for 22.5°, 67.5°, 112.5°, and 157.5° to improve the accuracy of the similarity measure, and its specific directions are shown in Figure 1.

Based on the way 0°, 45°, 90°, and 135° were defined, the gradient equations for the other four directions (the dashed arrow directions in Figure 1) were defined as follows:

$$d_{\text{grad22.5}}(x,y) = \sum_{i=0}^{k}\sum_{j=0}^{k}\left(I(x+i,y+j) - I(x+i+2,y+j+1)\right)^2, \tag{10}$$

$$d_{\text{grad67.5}}(x,y) = \sum_{i=0}^{k}\sum_{j=0}^{k}\left(I(x+i,y+j) - I(x+i+1,y+j+2)\right)^2, \tag{11}$$

$$d_{\text{grad112.5}}(x,y) = \sum_{i=0}^{k}\sum_{j=0}^{k}\left(I(x+i,y+j) - I(x+i-1,y+j+2)\right)^2, \tag{12}$$

$$d_{\text{grad135}}(x,y) = \sum_{i=0}^{k}\sum_{j=0}^{k}\left(I(x+i,y+j) - I(x+i-2,y+j+1)\right)^2. \tag{13}$$

Here (x, y) represents the coordinates of any pixel point in the noise-containing image. k was the size of the neighborhood, and $g(x, y)$ represents the gray value of the pixel point with (x, y) as the coordinates.

Based on the above definition, then the new gradient feature formula for pixel point (x, y) can be defined as follows:

$$\begin{aligned}\text{TD}_{\text{grad}(x,y)} = \min\{&d_{\text{grad0}}(x,y), d_{\text{grad22.5}}(x,y), d_{\text{grad45}}(x,y),\\ &d_{\text{grad67.5}}(x,y), d_{\text{grad90}}(x,y), d_{\text{grad112.5}}(x,y),\\ &d_{\text{grad135}}(x,y), d_{\text{grad157.5}}(x,y)\}.\end{aligned} \tag{14}$$

Gradient features reflect changes in the brightness of neighborhood pixel gray values, and spatial features were used to measure changes in detail. Therefore, the neighborhood multifeatures of a pixel point (x, y) can be characterized by combining the pixel grayscale feature, the gradient feature, and the spatial feature as follows:

$$T(x,y) = \min\left(\text{TD}_{\text{grad}}(x,y) \quad d_{\text{space}}(x,y)\, d_{\text{gray}}(x,y)\right). \tag{15}$$

Here $d_{\text{space}}(x, y)$ represents spatial similarity features and $d_{\text{gray}}(x, y)$ represents grayscale similarity features, the formula for multifeature similarity measure between the neighborhoods of two-pixel points (x_1, y_1) and (x_2, y_2) can be expressed as follows:

$$d_{\text{mul}} = \left\| T(x_1,y_1) - T(x_2,y_2) \right\|. \tag{16}$$

3.3. Dual Kernel Function and Smoothing Parameter Adaptive Computation. In high-intensity noise, the Gaussian kernel function had poor denoising performance but maintained good detail, while the sinusoidal kernel function had good denoising ability but tended to overweight the results, resulting in blurred denoising. To combine the advantages of both Gaussian kernel functions and sinusoidal kernel functions, the improved kernel function can be expressed as follows:

$$W_{(x,y)} = \begin{cases} \exp\left(\dfrac{d_{\text{mul}}^2}{2(h_1)^2}\right) * \left(\sin\left(\dfrac{\pi d_{\text{mul}}}{h_1}\right)/\pi d_{\text{mul}} h_2\right) & 0 < d_{\text{mul}} < h_1 \\ 0 & \text{others} \end{cases}. \tag{17}$$

Here h_1, h_2 represent different values of the smoothing parameter and d_{mul} represents the multifeature similarity between neighborhoods.

The dual kernel function can effectively avoid the interference of less similar neighborhoods and achieve a balance between the Euclidean distance and similarity weights, thus effectively enhancing the denoising effect and improving image detail protection. In image smoothing regions, where pixel redundancy was greater, a larger h_1 value could improve the degree of smoothing; in edge/texture regions, where pixel redundancy was less, a smaller h_1 value was good for maintaining edges, so the smoothing parameters could be selected adaptively according to the image region classification results. Based on the approximate relationship between the filtering parameter h_1 and the noise variance under the Gaussian distribution characteristics, the optimal filtering parameters for

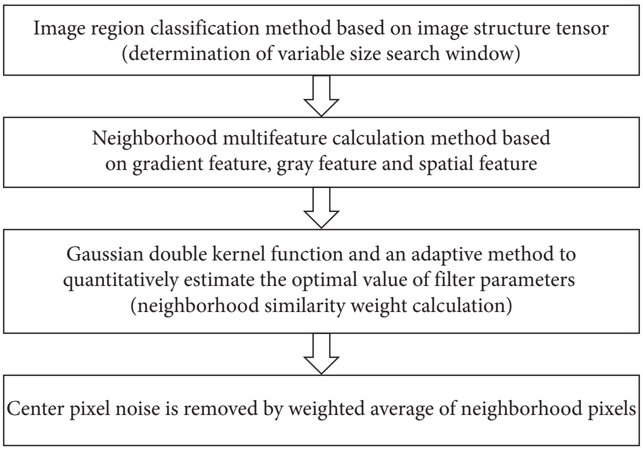


Image region classification method based on image structure tensor (determination of variable size search window)
Neighborhood multifeature calculation method based on gradient feature, gray feature and spatial feature
Gaussian double kernel function and an adaptive method to quantitatively estimate the optimal value of filter parameters (neighborhood similarity weight calculation)
Center pixel noise is removed by weighted average of neighborhood pixels

Figure 2: The overall technology diagram.

the three different types of regions for images with a gray level of 255 can be defined as follows:

$$h_1 = \begin{cases} \left[2\sqrt{2} + \dfrac{\max(0, (\sigma_1 - 70))}{10}\right]\sigma^2, & I(x,y) \in S1 \\ \left[2.33\sqrt{2} + \dfrac{\max(0, (\sigma_1 - 70))}{10}\right]\sigma^2, & I(x,y) \in S2\,. \\ \left[3\sqrt{2} + \dfrac{\max(0, (\sigma_1 - 70))}{10}\right]\sigma^2, & I(x,y) \in S3 \end{cases} \tag{18}$$

Here σ represents the noise variance and σ_1 represents the standard deviation of the noisy image, $S1$ represents the edge/texture region, $S2$ represents the transition region, and $S3$ represents the smoothed region.

3.4. Image Denoising. Determining the neighborhood search window size and the smoothing filtering parameters, the improved NLM formulation in this paper was as follows:

$$\overline{N}(x) = \frac{1}{\sum\limits_{y\in S} W(x,y)} \sum_{y\in S} W(x,y)N(y). \tag{19}$$

Here $\overline{N}(x)$ represents the filtered gray value of pixel point x in the noise-containing image, $N(y)$ represents the gray value at point y of the noise-containing image, S_x denoted the coordinate domain of the pixel point in the search neighborhood centered at x, and $W(x,y)$ represent the dual kernel function.

In summary, the block diagram of the algorithmic scheme proposed in this paper is shown in Figure 2.

The steps can be summarized as follows:

Step 1: Inputting noise-containing images;

Step 2: Use Equation (9) to implement region division and determine the search window size for different regions; use Equation (18) to calculate different regional filtering parameters;

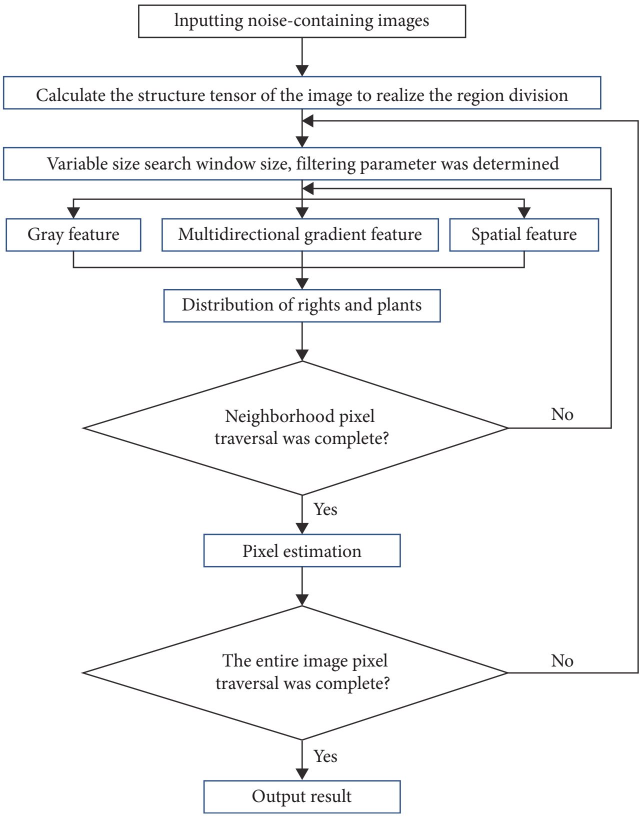

Inputting noise-containing images
Calculate the structure tensor of the image to realize the region division
Variable size search window size, filtering parameter was determined
Gray feature
Multidirectional gradient feature
Spatial feature
Distribution of rights and plants
Neighborhood pixel traversal was complete?
No
Yes
Pixel estimation
The entire image pixel traversal was complete?
No
Yes
Output result

Figure 3: Flowchart of multifeature NLM algorithm with variable size windows.

Step 3: Use Equation (15) to calculate the multifeatures between neighbors;

Step 4: Calculation of similarity neighborhood weights using Equations (16) and (17);

Step 5: Normalizing the processing weight values and traversing the neighborhood pixels for steps (3) and (4);

Step 6: Noise removal using Equation (19) for the target pixels;

Step 7: Applying steps (3)–(6) to the entire image, down to the last pixel;

Step 8: Output the resulting image.

The flowchart of the algorithm is shown in Figure 3

For a noisy image of size $\sqrt{N} \times \sqrt{N}$, the computational complexity of the algorithm for denoising using this algorithm was $O(N \times P \times M)$ which assumes that the size of the neighborhood used was $\sqrt{P} \times \sqrt{P}$, and the size of the search window was $\sqrt{M} \times \sqrt{M}$.

4. Algorithm Testing and Application

4.1. Experimental Setting. The equipment configuration for this experiment was Intel(R) Core(TM) I5-7500 with 8.00 GB of RAM, Windows 10, and the processing software was MATLAB R2022b. Eight standard grayscale images (Jetplane 512 × 512, Boat 512 × 512, Lena 512 × 512, Parrot 512 × 512, Barbara 512 × 512, C.man 256 × 256, Lax 256 × 256, Peppers 256 × 256) shown in Figure 4 were used as the test image set,

Figure 4: Test image set (noiseless image): (a) jetplane; (b) C.man; (c) boat; (d) lax; (e) lena; (f) peppers; (g) parrot; (h) barbara.

which were added to additive Gaussian white noise with mean 0 and different noise standard deviations σ. They were tested by the traditional NLM method, the Gaussian kernel function NLM method, literature [12–15], and the method of this paper. For the convenience of analysis and evaluation, the conventional NLM algorithm and Gaussian kernel function NLM algorithm, literature [12–14] used a neighborhood window size of 9×9 and a search window size of 15×15 in successive simulation experiments and literature [15] used a neighborhood window size of 9×9 and a search window size of 9×9, with several iterations of 3 and a step size of 6. The algorithm in this paper had a neighborhood window size of 9×9 and a search window size of 9×9 (edge/texture region), 15×15 (transition region), and 21×21 (smoothing region) for the three sizes, respectively.

4.2. Determination of the Parameters. As shown in Equation (17), the filtering parameters affected the denoising

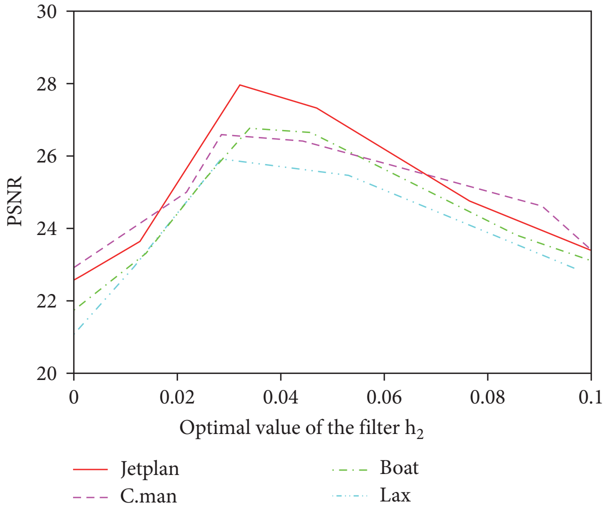


Figure 5: Schematic diagram of the filtering parameters and PSNR variation curve.

Table 1: PSNR (dB) table for different simulation results.

Image	σ	Noisy image	Traditional NLM	Gaussian kernel NLM	Literature [12] method	Literature [13] method	Literature [14] method	Literature [15] method	Algorithms in this paper
Jetplane	20	22.57	28.12	28.45	28.77	29.25	29.56	30.45	30.38
	40	17.26	26.32	26.65	26.92	27.25	27.52	28.15	28.57
	60	15.37	23.12	24.45	24.72	25.25	25.42	25.75	26.81
C.man	20	22.68	25.45	26.88	26.95	27.28	28.45	28.88	28.72
	40	17.39	23.21	23.52	24.31	24.62	25.61	26.32	26.88
	60	15.48	22.47	22.61	23.37	23.71	24.47	24.71	25.14
Boat	20	22.32	27.21	27.78	28.21	28.78	29.21	29.78	28.62
	40	17.29	24.21	24.83	25.21	25.83	26.21	26.83	27.56
	60	15.61	22.73	23.32	23.73	24.32	24.73	25.32	25.58
Lax	20	22.64	24.12	24.63	25.12	25.83	26.72	27.23	27.06
	40	17.43	23.21	23.72	24.21	24.82	25.21	25.98	26.42
	60	15.26	20.21	21.63	22.41	22.73	23.21	23.83	24.12
Lena	20	22.49	29.06	29.32	29.75	29.98	30.04	30.13	31.15
	40	17.98	24.03	24.67	25.31	25.92	26.36	26.89	26.69
	60	15.67	22.23	23.28	23.85	24.36	24.82	25.27	25.84
Peppers	20	22.71	28.96	29.18	29.69	29.75	30.15	30.42	30.12
	40	17.42	24.15	24.36	25.27	25.74	26.18	26.91	27.82
	60	15.26	23.01	23.41	23.26	24.29	24.75	25.31	25.61

results; therefore, the optimal parameters needed to be determined. In this paper, the filtering parameter with the maximum peak signal-to-noise ratio (PSNR) value was selected as the optimal parameter. The h_1 was calculated adaptively using Equation (18); thus only the optimal value of h_2 needed to be determined. When testing h_2, the value of h_1 was fixed at 0.02, and the value of h_2 was varied from 0 to 0.1, and then the optimal value of the filtering parameter h_2 was determined according to the PSNR variation curve of the test image. Four grayscale images (jetplane, C.man, boat, lax) shown in Figure 4 were used for simulation tests. The images were added with Gaussian white noise with mean $\mu = 0$ and standard deviation $\sigma = 40$, then the relationship between h_2 and PSNR changes as shown in Figure 5.

From Figure 5, it can be seen that the PSNR transformation trend of the test image was increasing and then decreasing, and the best denoising effect was achieved when the PSNR value was maximum at $h_2 = 0.03$. Therefore, h_2 was tested with the above value in the subsequent simulation.

4.3. Experimental Results and Their Comparative Analysis. To analyze the effect of noise intensity on different denoising methods, Gaussian white noise with mean $\mu = 0$ and standard deviation$\sigma = 20, 40, 60$ was added to the six standard grayscale images shown in Figure 4, and then the denoising process was tested by using seven methods. The values of PSNR and structural similarity (SSIM) for 126 simulation results are shown in Tables 1 and 2, respectively.

Table 2: SSIM table for different simulation results.

Image	σ	Noisy image	Traditional NLM	Gaussian kernel NLM	Literature [12] method	Literature [13] method	Literature [14] method	Literature [15] method	Algorithms in this paper
jetplane	20	0.637	0.804	0.851	0.804	0.851	0.804	0.851	0.832
	40	0.601	0.761	0.768	0.771	0.781	0.791	0.799	0.815
	60	0.572	0.623	0.675	0.683	0.715	0.723	0.745	0.781
C.man	20	0.613	0.685	0.728	0.785	0.798	0.815	0.828	0.816
	40	0.581	0.628	0.631	0.648	0.651	0.678	0.731	0.768
	60	0.532	0.605	0.621	0.635	0.639	0.655	0.661	0.679
Boat	20	0.623	0.704	0.725	0.734	0.765	0.784	0.825	0.814
	40	0.564	0.631	0.652	0.671	0.692	0.711	0.752	0.792
	60	0.528	0.611	0.626	0.631	0.656	0.661	0.676	0.687
Lax	20	0.625	0.636	0.662	0.686	0.712	0.736	0.762	0.751
	40	0.579	0.629	0.648	0.659	0.668	0.689	0.718	0.732
	60	0.531	0.579	0.611	0.639	0.641	0.651	0.661	0.665
Lena	20	0.612	0.713	0.732	0.748	0.772	0.786	0.827	0.821
	40	0.599	0.635	0.661	0.679	0.702	0.718	0.749	0.789
	60	0.524	0.612	0.624	0.641	0.648	0.657	0.671	0.692
Pepper	20	0.619	0.698	0.718	0.729	0.771	0.792	0.831	0.824
	40	0.576	0.641	0.652	0.661	0.672	0.686	0.724	0.774
	60	0.511	0.603	0.621	0.638	0.652	0.667	0.681	0.698

From Tables 1 and 2, it can be seen that at low noise ($\sigma = 20$), literature [15] obtained a better denoising effect with maximum PSNR and SSIM values. The PSNR and SSIM of the improved algorithm in this paper were closest to the maximum value, although not the maximum value. However, as the noise level increased ($\sigma = 40, \sigma = 60$), the method in this paper then obtained a better denoising effect, and its PSNR and SSIM were better than the comparative literature methods. In this case, at $\sigma = 40$, the PSNR value of the traditional NLM method was increased by an average of 38.571%, and the SSIM value reached an average of 0.654. The PSNR value of the Gaussian kernel function NLM results was increased by an average of 41.07%, and the SSIM value reached an average of 0.669. The PSNR value of literature [12] method results were increased by an average of 44.38%, and the SSIM value reached an average of 0.682. The PSNR values of the results of the method of literature [13] increased on average by 47.19%, and the SSIM values reached an average of 0.694. The PSNR values of the results of the method of literature [14] increased on average by 49.97%, and the SSIM values reached an average of 0.712. The PSNR values of the results of the method of literature [15] increased on average by 53.78%, and the SSIM values reached an average of 0.745. The PSNR values of the results of the method in this paper were improved by 56.53% on average, and the SSIM values reached 0.778 on average, as shown in Figures 6 and 7.

From the above experimental data, it can be seen that with the increase of noise intensity, the results of this paper's method have been maintaining higher values of PSNR and SSIM, indicating that this paper's algorithm had a better denoising ability and can maintain better image quality while removing noise.

To have a visual impression of the denoising effect of each algorithm on the noisy images, the denoising results of four noise-containing images with noise mean $\mu = 0$ and standard deviation $\sigma = 40$ were shown. Noise-containing images and denoising results are shown in Figures 8–11.

From Figures 8 to 11, it can be seen that although the traditional NLM algorithm removed the noise contained in the image, the excessive smoothing led to the loss of some detail information in the image, resulting in blurring of the details and the regions with rich edge information, as shown in Figures 8(b), 9(b), 10(b), and 11(b). The denoising effect of the Gaussian kernel function NLM algorithm was relatively good, but the denoising results showed some "bar" textures, as shown in Figures 8(c), 9(c), 10(c), and 11(c), which, combined with the original noiseless image, indicated that the above textures were extraneous noise. Literature [12] method improved the quality of the results, but the layout of the "ringing" phenomenon, as shown in Figures 8(d), 9(d), 10(d), and 11(d), affected the quality of the image. The local details of the results of literature [13] method were relatively blurred, as shown in Figures 8(e), 9(e), 10(e), and 11(e). Literature [14] method resulted in poor denoising of regions with lower brightness and blurring of local details, as shown in Figures 8(f), 9(f), 10(f), and 11(f). Literature [15] and the results of the method in this paper had the best noise removal, the closest to the original noise-free image, and the details were kept the clearest, especially the letters in the jetplane image as well as the fuselage, hull and other positions in the boat image were more easily recognized.

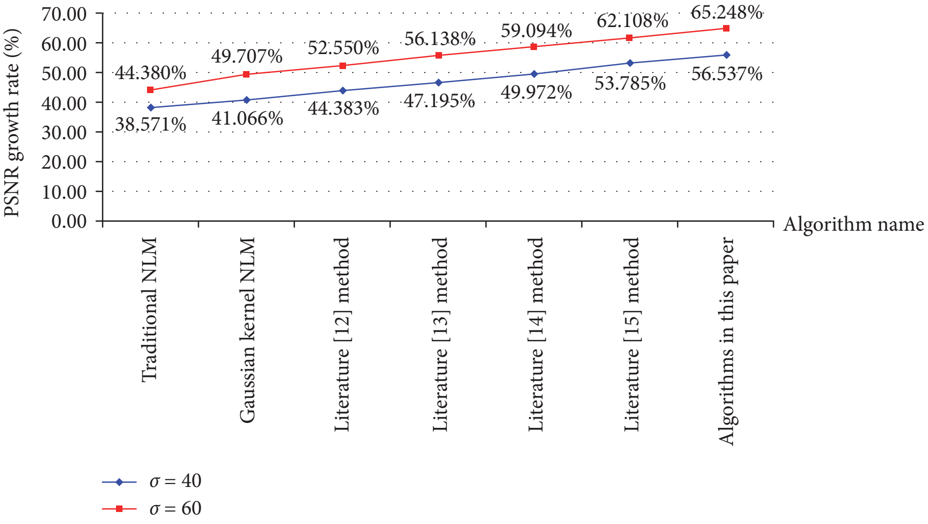


Figure 6: PSNR growth rate of different algorithms.

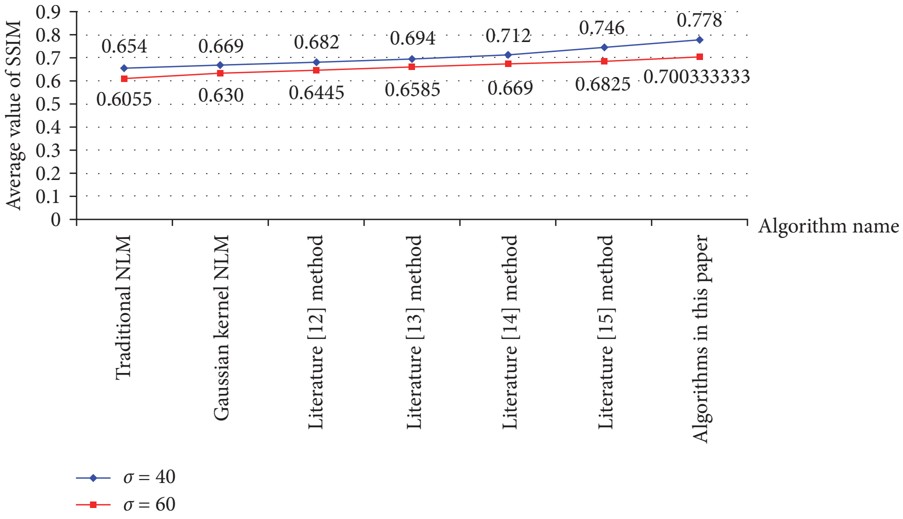


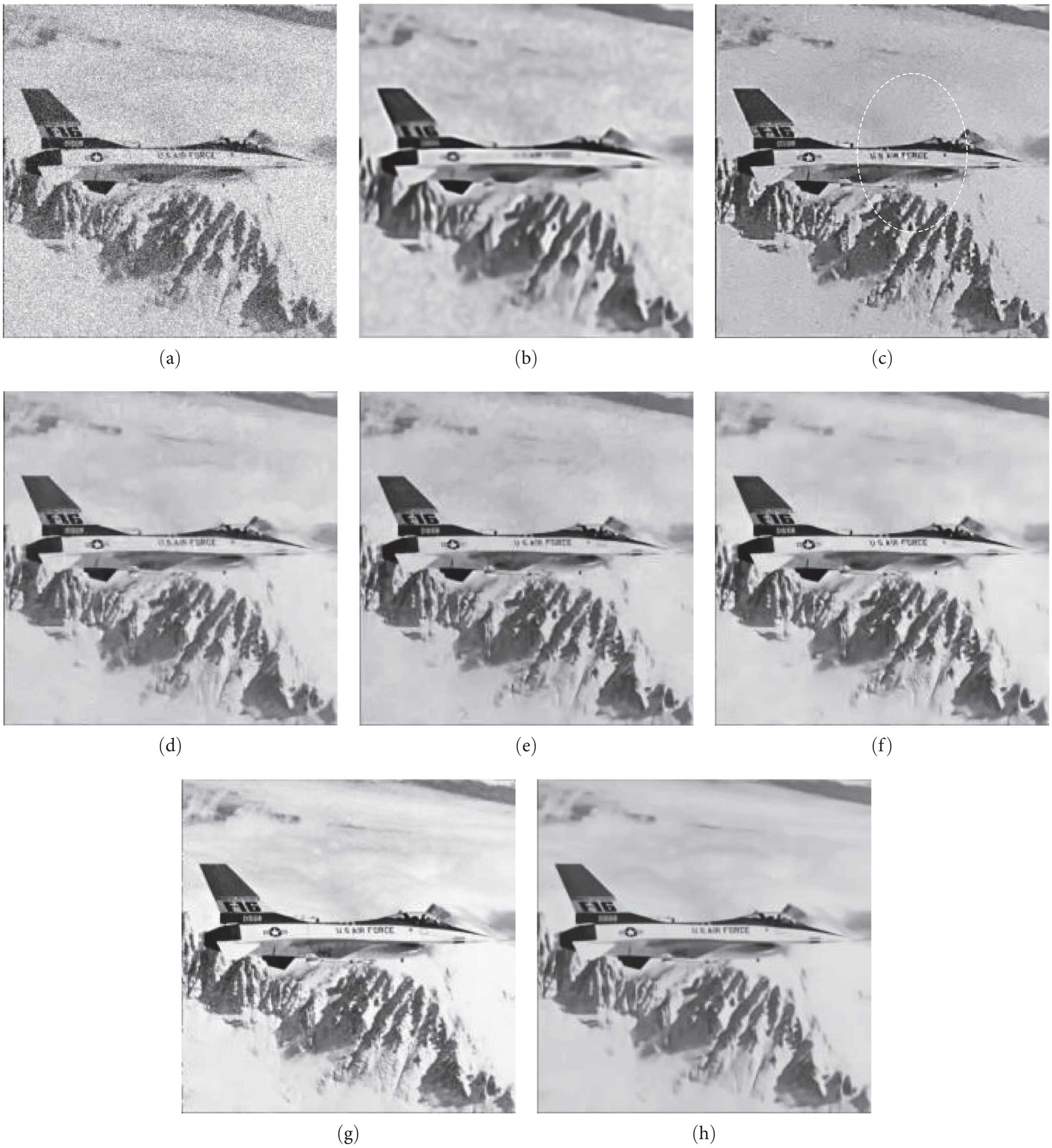
Figure 7: Average SSIM for different algorithms.

To further analyze the denoising effect of the methods in this paper, the noise residuals of the seven methods were displayed, as shown in Figures 12–15.

From Figures 12 to 15, it can be seen that the traditional NLM noise residuals contained more edge texture information, especially in Figures 12(a), 13(a), and 14(a), the contours of the aircraft, the camera frame, the hull of the ship can be seen, which showed that the algorithm lost a large amount of detail information while removing the noise. The Gaussian kernel function NLM algorithm removed the noise residuals with a similar "bar" texture, as shown by the wireframes in Figures 12(b), 13(b), 14(b), and 15(b). Detailed information such as edges/textures were also present to varying degrees in the noise residuals of literature [12–14] and in the noise residuals of literature [15]. In the noise residuals of this paper's method, the detail information, such as edge texture, was the least, and there were almost few signals that were continuous or had a certain morphology, indicating better protection of detail information, as shown in Figures 12(g), 13(g), 14(g), and 15(g).

To better evaluate the quality of the denoised image, the denoising results can be evaluated using a completely nonreferential algorithm (natural image quality evaluator, NIQE), and the expert evaluation values (in percentage) were added together to obtain the evaluation averages of the seven methods as shown in Table 3. As can be seen from Table 3, the denoised image obtained using the denoising algorithm proposed in this paper was of the highest quality and was able to retain the details and texture features of the image better.

Next, the time spent by different processing methods to process different images was counted, and the average time was derived using 30 experiments, respectively, as shown in Figure 16. From Figure 16, it can be seen that compared to other denoising methods, the running time of the method

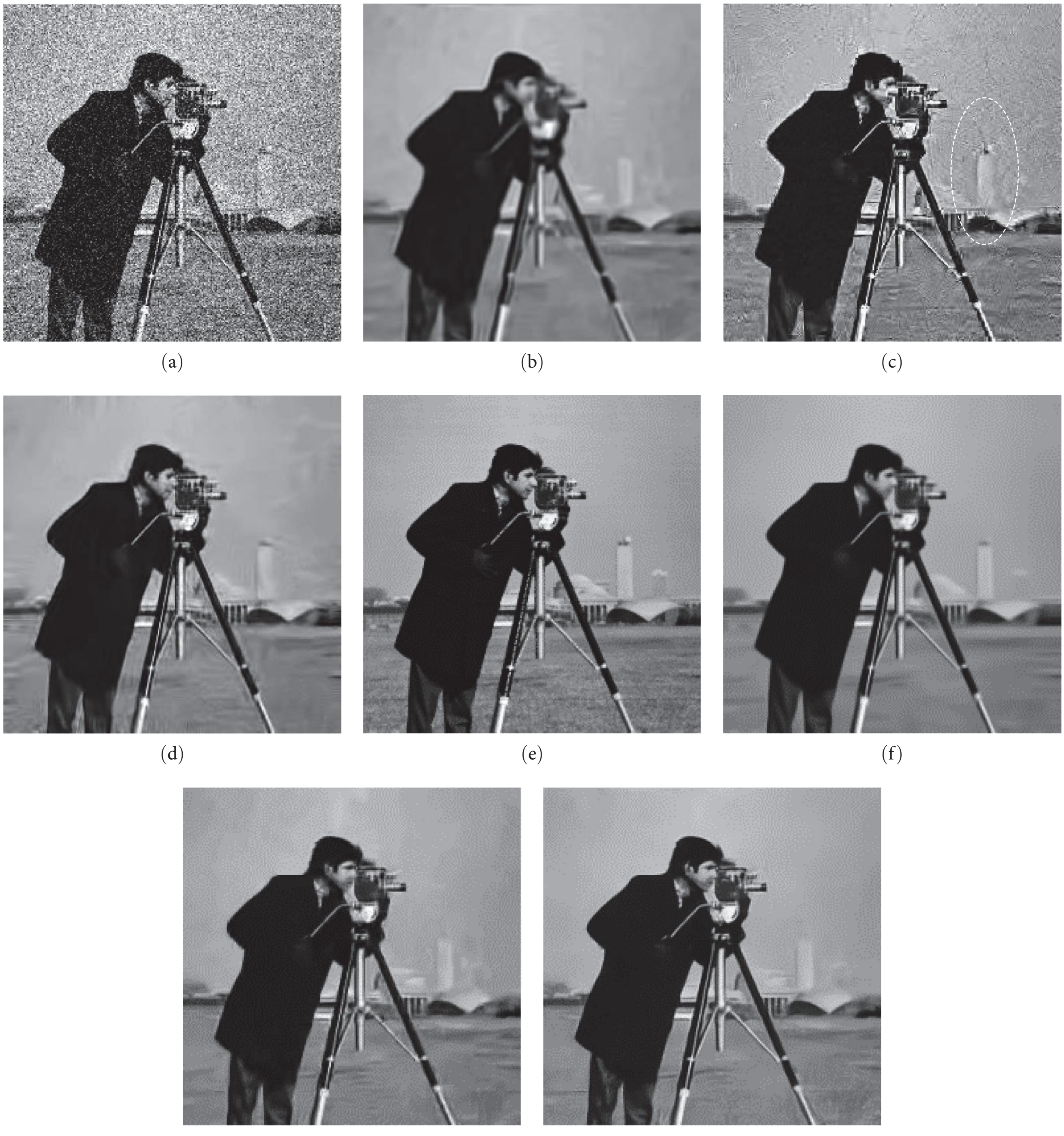


(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 8: Noisy images and denoising results of different algorithms (jetplane): (a) noise image; (b) traditional NLM denoising results; (c) Gaussian kernel NLM denoising results; (d) literature [12] denoising results; (e) literature [13] denoising results; (f) literature [14] denoising results; (g) literature [15] denoising results; (h) denoising results of this paper's method.



(a)

(b)

(c)

(d)

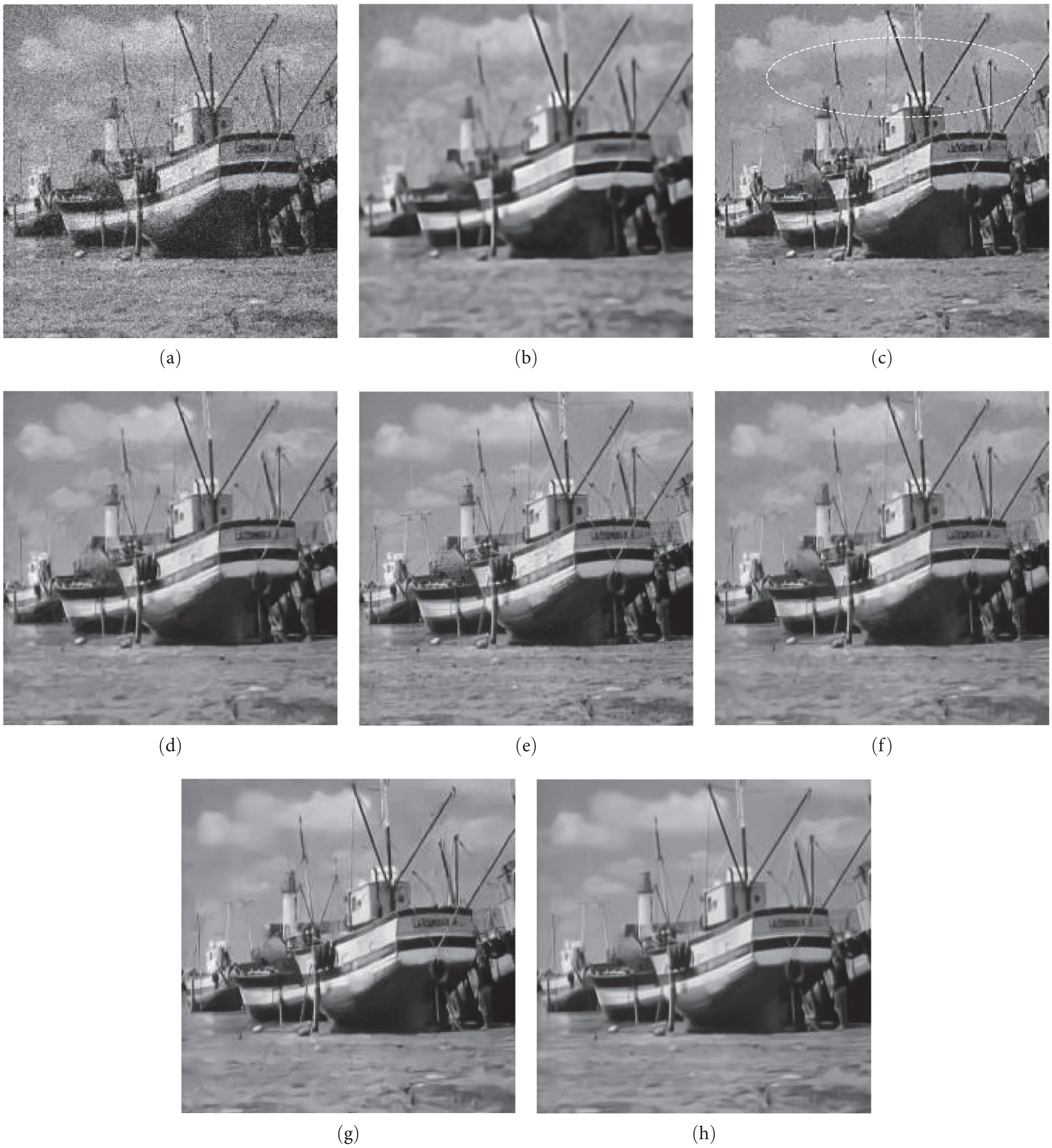
(e)

(f)

(g)

(h)

Figure 9: Noisy images and denoising results of different algorithms (C.man): (a) noise image; (b) traditional NLM denoising results; (c) Gaussian kernel NLM denoising results; (d) literature [12] denoising results; (e) literature [13] denoising results; (f) literature [14] denoising results; (g) literature [15] denoising results; (h) denoising results of this paper's method.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 10: Noisy images and denoising results of different algorithms (boat): (a) noise image; (b) traditional NLM denoising results; (c) Gaussian kernel NLM denoising results; (d) literature [12] denoising results; (e) literature [13] denoising results; (f) literature [14] denoising results; (g) literature [15] denoising results; (h) denoising results of this paper's method.

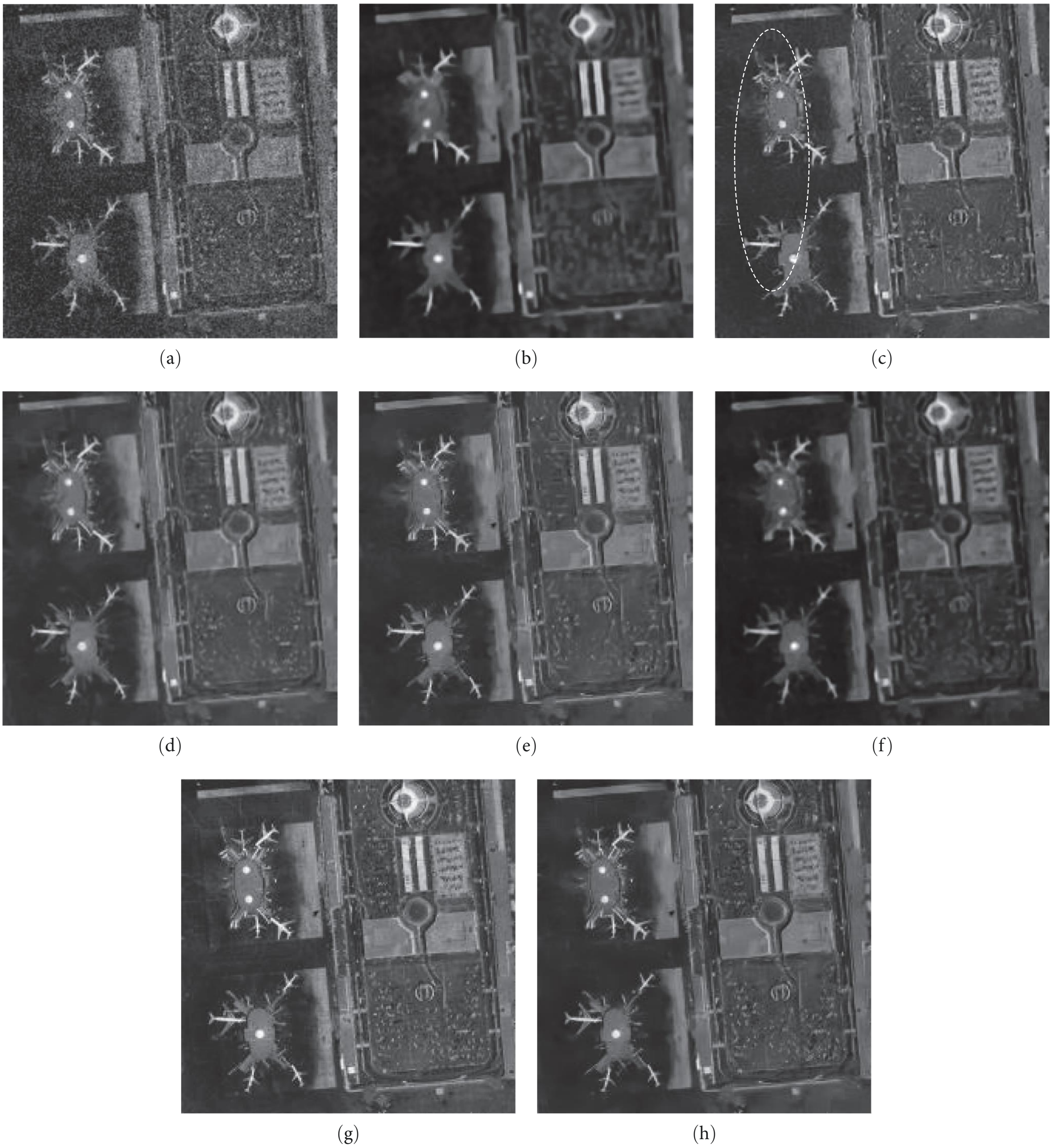
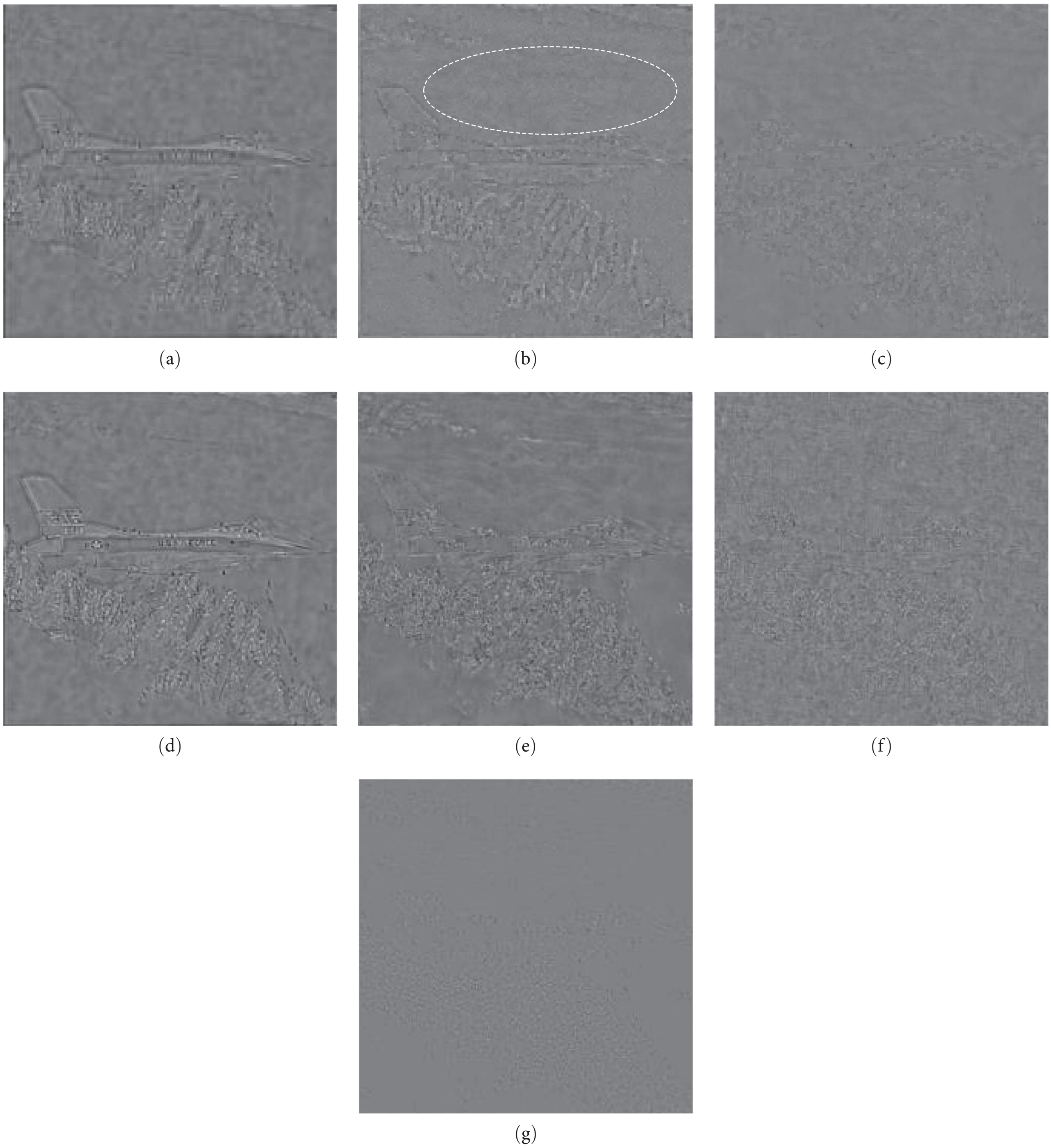


Figure 11: Noisy images and denoising results of different algorithms (lax): (a) noise image; (b) traditional NLM denoising results; (c) Gaussian kernel NLM denoising results; (d) literature [12] denoising results; (e) literature [13] denoising results; (f) literature [14] denoising results; (g) literature [15] denoising results; (h) denoising results of this paper's method.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 12: Noise residuals of test image (jetplane): (a) traditional NLM method residuals; (b) Gaussian kernel NLM method residuals; (c) literature [12] method residuals; (d) literature [13] method residuals; (e) literature [14] method residuals; (f) literature [15] method residuals; (g) residuals of the method in this paper.

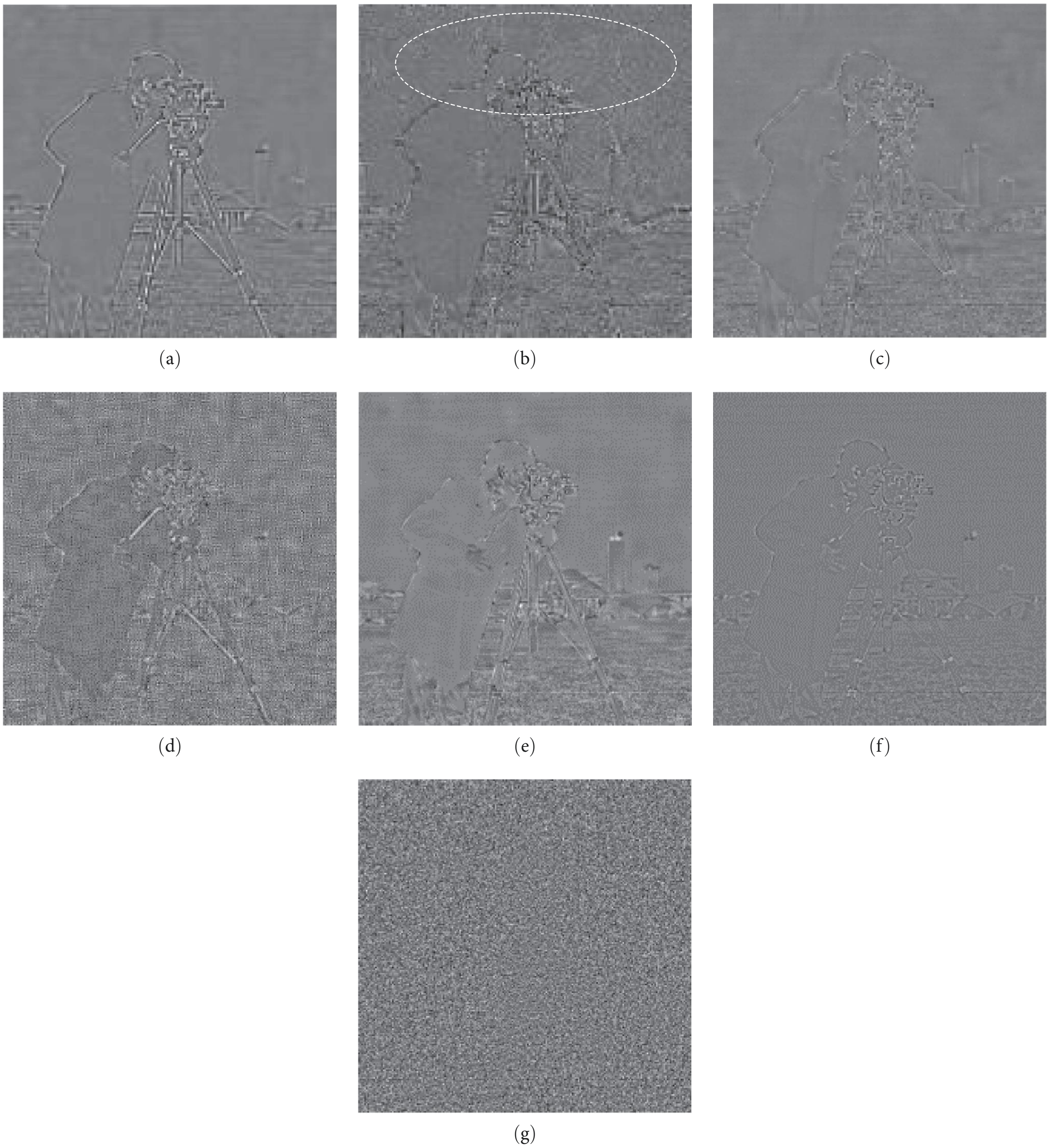
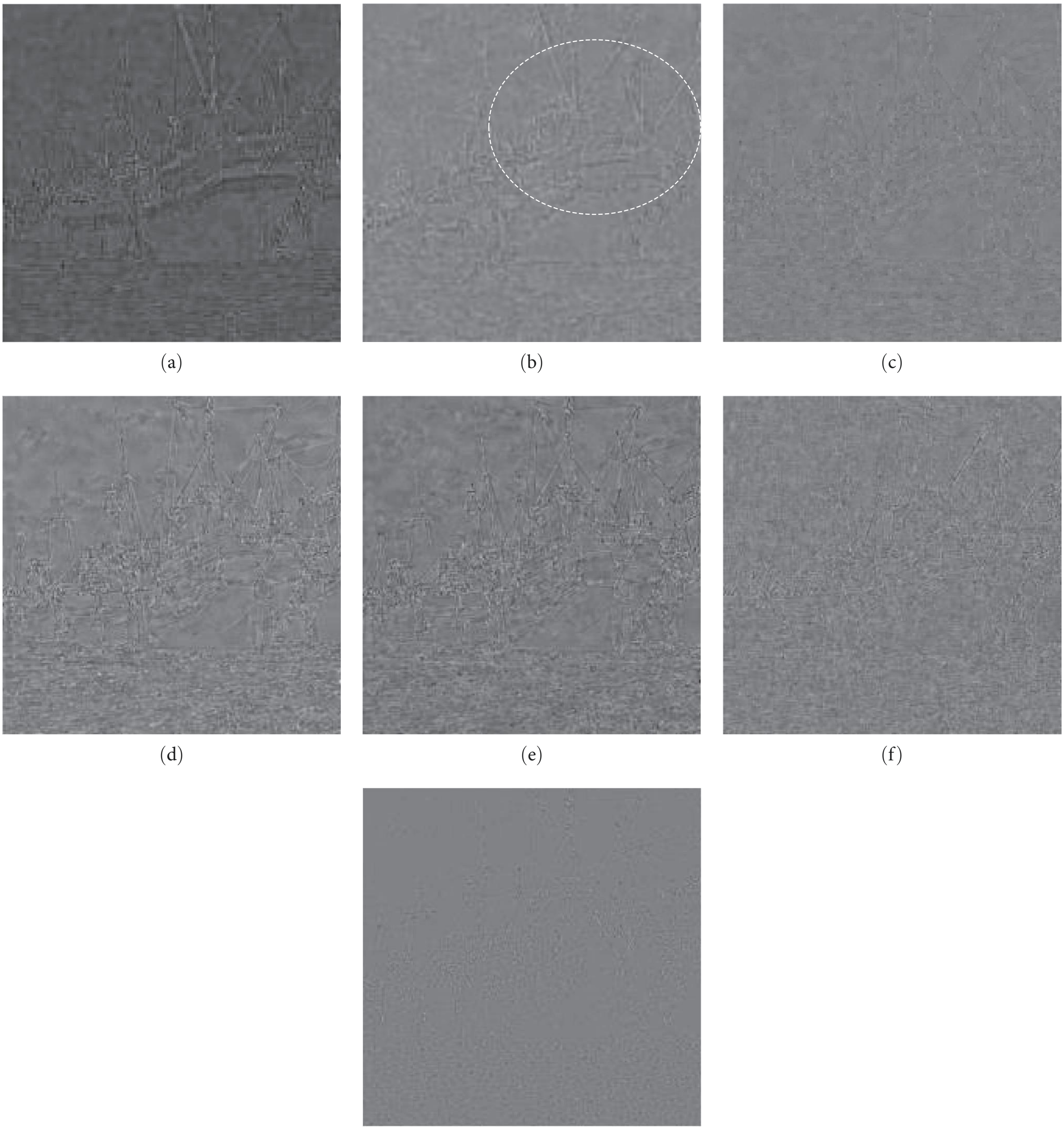


Figure 13: Noise residuals of test image (C.man): (a) traditional NLM method residuals; (b) Gaussian kernel NLM method residuals; (c) literature [12] method residuals; (d) literature [13] method residuals; (e) literature [14] method residuals; (f) literature [15] method residuals; (g) residuals of the method in this paper.



(a)

(b)

(c)

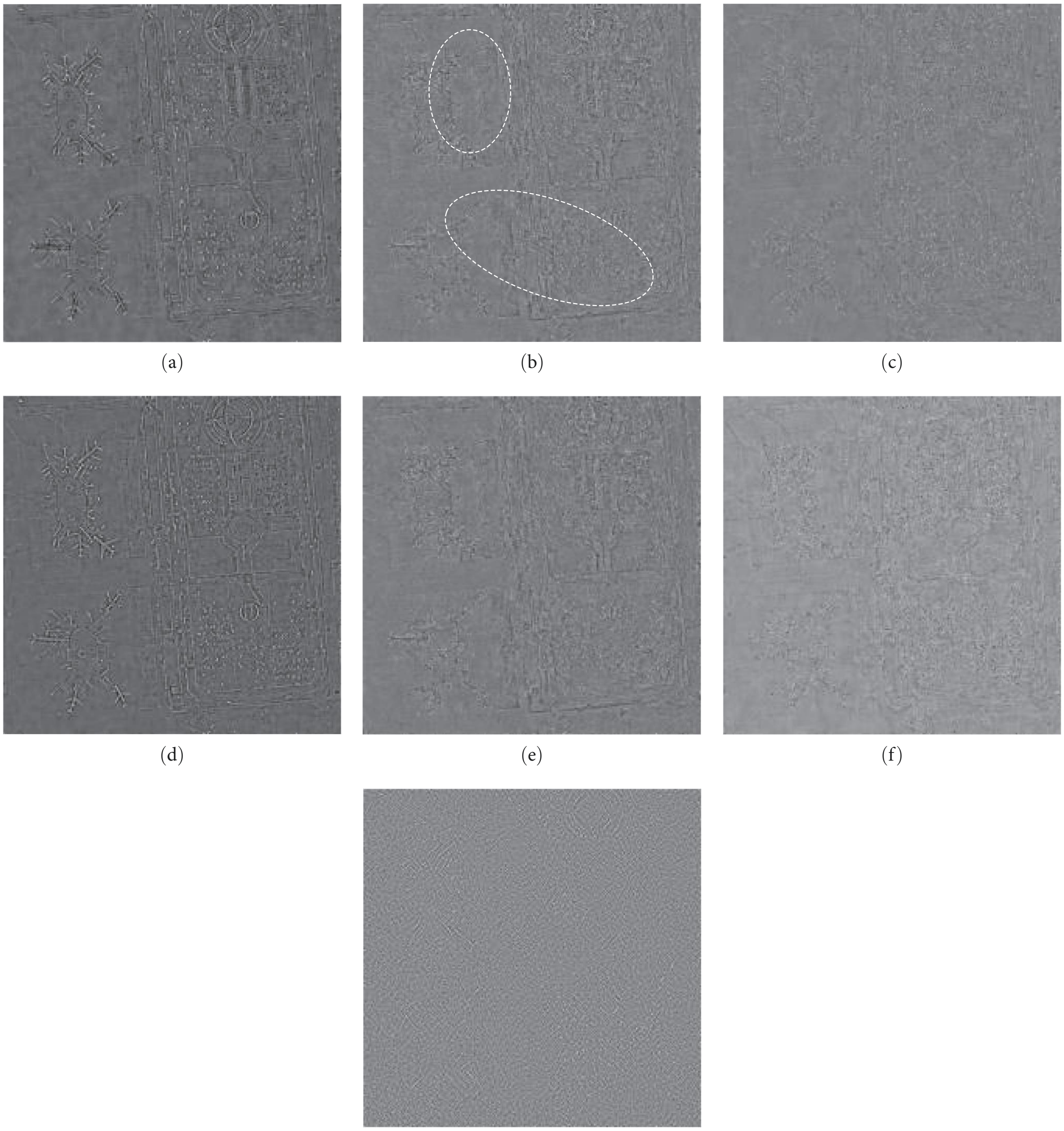
(d)

(e)

(f)

(g)

Figure 14: Noise residuals of test image (boat): (a) traditional NLM method residuals; (b) Gaussian kernel NLM method residuals; (c) literature [12] method residuals; (d) literature [13] method residuals; (e) literature [14] method residuals; (f) literature [15] method residuals; (g) residuals of the method in this paper.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 15: Noise residuals of test image (lax): (a) traditional NLM method residuals; (b) Gaussian kernel NLM method residuals; (c) literature [12] method residuals; (d) literature [13] method residuals; (e) literature [14] method residuals; (f) literature [15] method residuals; (g) residuals of the method in this paper.

Table 3: Evaluation value of a denoise method.

Denoise method	NIQE value	Expert rated values
Original noisy image	5.72	67
Traditional NLM	6.75	72
Gaussian kernel NLM	6.85	75
Literature [12] method	7.46	81
Literature [13] method	7.57	83
Literature [14] method	7.38	80
Literature [15] method	7.61	84
Algorithms in this paper	7.82	87

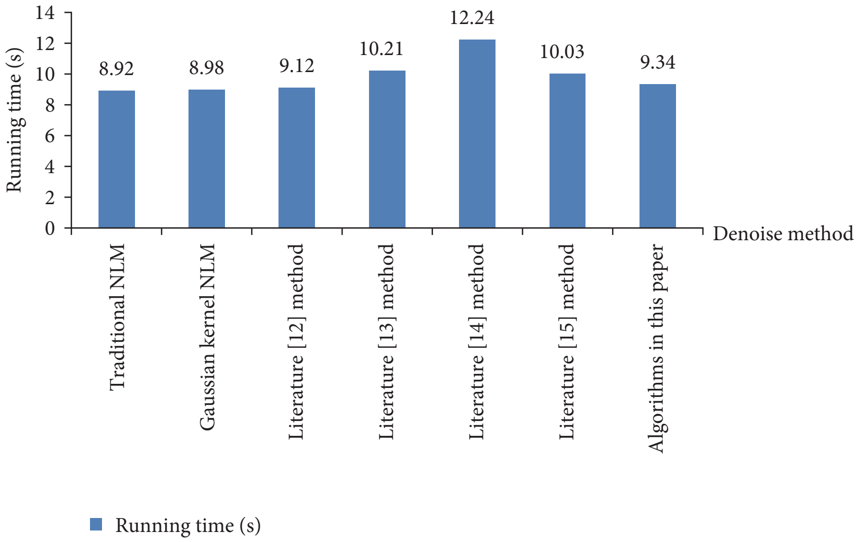


Figure 16: Running time statistics of different denoising methods.

proposed in this paper was not the shortest but still better than the methods proposed in literature [13–15].

5. Summary

In this paper, a dual-kernel adaptive NLM algorithm based on the variable-size window and neighborhood multifeature was used for digital image noise removal, which could effectively reduce the interference of neighborhood noise on similarity computation, thus improving the accuracy of neighborhood pixel similarity computation. The algorithm applied a variable-size search window to determine the search window size and smoothing parameters of the current neighborhood and adapted the Gaussian–Sinusoidal dual-kernel function for similarity weight calculation, which effectively improved the denoising effect and the protection of edge texture based on effectively improving the image quality and signal-to-noise ratio. After simulation and verification, the method in this paper had a better denoising effect and edge/texture protection ability, which had a certain value of popularization and application in digital image processing. However, the experimental images used in this study were the standard test images in the field of image processing. How to extend this research to hyperspectral images and achieve real-time processing results in a short period will be a pressing issue to be addressed in the next research. Meanwhile, how to combine this algorithm with the latest research techniques, such as sparse representation and neural networks, to obtain more stable and better denoising results is also an area that needs to be further worked on in future work.

Data Availability

All the data used to support the findings of the study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank the professors and students at the University of Kitakyushu for their generous help with the full paper, from the investigation to the checking process. This study was supported by the performance-based salary distribution system in local colleges and universities, project no.: 21JK0007; Teaching Reform of Software Engineering Course under the Background of New Engineering and Engineering Certification, project no.:

JG202225; Ankang City Science and Technology Plan Project in 2023 (project name: Research on Mixed Noise and Its Removal Algorithm in Digital Images).

References

[1] Z. Yiting, C. Lei, and Y. Yanying, “Image smoothing algorithm based on L2,1 norm regularization matrix decomposition,” *Journal of Nanchang University (Natural Science)*, vol. 39, no. 5, pp. 426–431, 2015.

[2] H. Xiaowei, S. Yi, and W. Xiao, “Cerenkov fluorescence image denoising algorithm based on spatial information improved clustering,” *Acta Optica Sinica*, vol. 38, no. 10, pp. 259–267, 2018.

[3] S. Dangguo, D. Yangyang, and X. Yan, “Ultrasonic speckle noise reduction based on adaptive Gaussian filter,” *Journal of Data Acquisition & Processing*, vol. 32, no. 4, pp. 746–753, 2017.

[4] Z. Jianli and S. Shenshen, “Ore image segmentation based on genetic algorithm optimal threshold segmentation,” *Science Technology and Engineering*, vol. 19, no. 7, pp. 105–109, 2019.

[5] H.-H. Kang, D.-H. Shin, and E.-S. Kim, “Compression scheme of sub-images using Karhunen–Loeve transform in three-dimensional integral imaging,” *Optics Communications*, vol. 281, no. 14, pp. 3640–3647, 2008.

[6] L. Hong, H. Jidong, and F. Yi, “A review of active contour models in medical image segmentation,” *Science Technology and Engineering*, vol. 18, no. 16, pp. 161–167, 2018.

[7] J. Hongbin, *Research on 3D Filtering and Block Matching Algorithm Based on Wavelet Transform*, Dalian Maritime University, Dalian, 2018.

[8] C. Hongyan and M. A. Shang, “Application of independent component analysis in image processing,” *Computer Engineering*, vol. 37, no. S1, pp. 261–277, 2011.

[9] W. Dejiao, S. Jinfang, and W. Qian, “Application of integrative denoising method for radiation image denoising,” *Machinery Design & Manufacture*, vol. 311, no. 1, pp. 97–100, 2017.

[10] A. Buades, B. Coll, and J. M. Morel, “A non-local algorithm for image denoising,” in *IEEE Computer Society Conference on, IEEE*, pp. 60–65, 2005.

[11] R. Vignesh, B. T. Oh, and C. C. J. Kuo, “Fast non-local means computation with probabilistic early termination,” *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 277–280, 2010.

[12] H. Yuzheng, Y. Cihui, and L. Quan, “Dual kernel non-local means denoising algorithm based on gradient feature,” *Application Research of Computers*, vol. 36, no. 5, pp. 1573–1576, 2019.

[13] L. Haiqing and G. Hongwei, “Two-stage non-local means denoising based on hybrid robust weight and improved method noise,” *Computer Engineering & Science*, vol. 40, no. 7, pp. 1227–1236, 2018.

[14] Y. Han, Z. Xuxiu, and D. Mingyan, “An improved non-local means denoising algorithm,” *Electronic Measurement Technology*, vol. 42, no. 22, pp. 145–149, 2019.

[15] L. Zhuhua and C. Lingyan, “Detail preserving interactive non-local means algorithm for de-speckling of ultrasound images,” *Computer Engineering and Design*, vol. 41, no. 2, pp. 477–482, 2020.

[16] F.-E. Shervan, S. Fakhrahmad, and F. Tajeripour, “Impulse noise reduction for texture images using real word spelling correction algorithm and local binary patterns,” *The International Arab Journal of Information Technology*, vol. 15, no. 6, pp. 1024–1030, 2018.

[17] K. Asem, “Digital image noise removal based on collaborative filtering approach and singular value decomposition,” *Multimedia Tools and Applications*, vol. 12, no. 81, pp. 16645–16660, 2022.

[18] A. Khmag, “Additive Gaussian noise removal based on generative adversarial network model and semi-soft thresholding approach,” *Multimedia Tools and Applications*, vol. 82, pp. 1–12, 2022.

[19] A. Khmag, A. R. Ramli, S. A. R. Al-haddad, and N. Kamarudin, “Natural image noise level estimation based on local statistics for blind noise reduction,” *The Visual computer*, vol. 34, no. 4, pp. 575–587, 2018.

[20] X. Guangyu and I. Shexiang, “Non-local mean denoising based on fuzzy metric visual features,” *Computer Engineering and Science*, vol. 39, no. 6, pp. 1133–1140, 2017.

[21] L. Junrui, L. Xuegang, Q. Shifeng, and P. Zhenming, “Non-local mean denoising based on visual features of fuzzy measures,” *Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition)*, vol. 30, no. 3, pp. 408–415, 2018.