Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

Research article

An improved BM3D algorithm based on anisotropic diffusion equation

Yanyan Zhang[1,2,3,]*** and Jingjing Sun**[1]

[1] Jiangsu Key Laboratory of Meteorological Observation and Information Processing, Department of Electronic and Information Engineering, Nanjing University of Information & Science Technology, Nanjing 210044, China
[2] Jiangsu Technology & Engineering Center of Meteorological Sensor Network, Nanjing 210044, China
[3] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing 210044, China

* Correspondence: Email: 002243@nuist.edu.cn.

Abstract: Traditional 3D block matching (BM3D) algorithms are among the best denoising methods at present; however, they exhibit the issue of ringing around image edges, which makes them unable to protect image edges and details. Therefore, this paper proposes an BM3D noise processing algorithm for the diffusion equation to reduce image noise without affecting image details, specifically at the edges. This method first uses anisotropic diffusion (AD) filtering for image preprocessing, and then uses the edge direction instead of horizontal direction to search for similar blocks. The AD model is mainly improved to achieve better edges and detailed processing effects. Firstly, with the improved AD direction, a 5×5 edge enhancement operator model is implemented in eight directions, and the corresponding gradient information is obtained. This operator improves the processed image edges to achieve clear contours and good continuity. Next, a new calculation method for the diffusion function, whose coefficient is constructed using a hyperbolic tangent function, is introduced. The proposed method is based on the link between the image gradient and diffusion function, and it is mathematically proven that the diffusion function converges faster than the diffusion function of the model proposed by Perona and Malik. Experimental results indicate that the improved model can effectively retain the image edges and texture details, avoid edge ringing, and provide significant improvements in terms of the subjective visual effects and objective numerical indicators.

Keywords: anisotropic diffusion; diffusion function; edge detection; image gradient; 3D block matching algorithm

1. Introduction

A large amount of noise is produced during image acquisition and transmission. The reduction of such noise is critical in image processing. This noise processing directly determines the feasibility and accuracy of several parts of image processing, including image segmentation, image classification, feature extraction, and pattern recognition. The objective of image denoising is to reconstruct images that are degraded by noise corrosion to improve their quality and consequently improve the interpretation and extraction of data. At present, there are many types of image denoising algorithms. Commonly used algorithms include the mean filtering method [1], variational and partial differential equation (PDE) method [2], and wavelet transform threshold method [3]. These concepts are implemented in local neighborhoods. Buades et al. proposed a non-local mean noise reduction method [4]. Dabov et al. [5] proposed a three-dimensional block-matching collaborative filtering algorithm based on the non-local mean algorithm. This algorithm combines the advantages of spatial and frequency domain denoising and provides one of the highest levels of image denoising in terms of objective evaluation criteria such as peak signal-to-noise ratio (PSNR) and subjective visual quality. The disadvantage of the Wiener filtering method used in the collaborative filtering of the BM3D algorithm is that the amount of calculation is relatively large. Further, it is easy to generate a ringing effect, and important image features are often lost when images with rich details and complicated boundaries are processed. Several researchers have attempted to improve the BM3D algorithm [6–9]. In 2015, Zhong et al. [6] proposed different contraction functions for different norm constraints, making better use of the sparseness and non-local similarity of wavelet functions. In 2017, Wang et al. [9] proposed a method of estimating the noise intensity using total variation values, making the block size and similar distance between blocks adaptive to the noise intensity. Although these algorithms have achieved better image denoising effects than the original BM3D filtering algorithms, they do not adequately protect image details and blur the image edges.

The PDE method can more suitably address the trade-off between noise reduction and edge retention. The most typical method is the anisotropic diffusion (AD) equation [10] proposed by Perona and Malik (P-M). The denoising principle in this model involves constructing a diffusion function according to the gradient of images. There is a large gradient at the edge and a small gradient in the flat area to help filter image noise while retaining edge information. Although AD has made remarkable advancements in image filtering, it still possesses many shortcomings such as the staircase effect, ill-posedness of the P-M equation, and deficiencies of the edge stop function. Catte et al. [11] proposed a regularized P-M model that eliminated the shortcomings associated with the inability to filter large noise points. Gilboa et al. [12] proposed a forward–reverse AD model that effectively enhanced image edge information. AD filtering is an iterative process that relies on parameters such as the diffusion coefficients and number of iterations. Therefore, methods of optimizing these parameters and improving the denoising effects have been proposed [13–15]. In 2016, Tebini et al. [16] proposed a diffusion function derived from a hyperbolic tangent function. The convergence speed of the flow function is much greater than that of the diffusion function in the P-M model, which accelerates the denoising rate of the model and reduces the number of calculations required. Other researchers have proposed various AD methods [17–22] for denoising while preserving the image content and avoiding staircase effects.

According to the above analysis, on the basis of making full use of the P-M model, this paper presents for the first time a diffusion coefficient function based on the hyperbolic tangent function, using the gradient information of the eight neighborhood directions of the image to perform AD filtering. Thin lines, weak edges, and textures and details are effectively retained. Next, similar blocks in the vertical and edge direction are searched for to accomplish denoising. The denoised image effectively retains image details and avoids the edge ringing effect caused by the BM3D algorithm.

The remainder of this paper is organized as follows. Section II introduces the basic theory of the BM3D and AD algorithms, then briefly describes the improvements made to the AD model. Section III discusses the improvements made to the existing BM3D algorithm, combining the advantages of the AD and BM3D image denoising algorithms. It mainly describes the improvement of the AD model from the aspects of edge detection and diffusion function and presents a quantitative analysis of the superiority of the improved model. Section IV uses the PSNR and structural similarity (SSIM) [23] as objective indicators to quantify and compare the experimental simulation results and to compare the filtered results obtained using subjective vision. The results show that compared with the non-local methods (NLMs) and BM3D filtering methods, the proposed algorithm more efficiently filters image noise and maintains image details, and it effectively avoids edge ringing. Finally, Section V summarizes the conclusions and future research prospects.

2. Related work

2.1. Basic theory of BM3D image denoising algorithm

The BM3D algorithm is a 3D joint filtering algorithm based on the concept of the NLM algorithm proposed by Dabov et al. This method combines the advantages of spatial and transform domain filtering. The denoising effect achieved is ideal. Not only does this approach provide a very good subjective visual effect for the human eye, but it also performs extremely well in common denoising evaluation methods, such as those utilizing the PSNR of an image.

The traditional BM3D algorithm consists of two stages: the first stage involves basic estimation of the noisy image, and the second stage entails further improvement of the denoising performance. The basic estimation value from the first stage is used in the second stage of collaborative filtering using a prior model. Each phase includes three steps: grouping, collaborative filtering, and aggregation. Grouping is performed to find similar blocks and aggregate them into a three-dimensional array; collaborative filtering is conducted to three-dimensionally transform the formed three-dimensional array and to harden the noise by hard thresholding the coefficients in the transform domain. All of the images in the group are obtained by inverse transformation block estimates, and then these estimates are returned to their original positions. Finally, aggregation is performed to find the final estimate of the real image by determining the weighted average of the estimated local blocks that overlap each other.

The BM3D denoising algorithm is one of the most advanced algorithms, but there are also shortcomings. The denoising effect of the algorithm will be improved only when a large number of high-quality matching blocks are searched for in the reference block, and the characteristics of details such as image edges and textures are not completely accounted for, especially when processing edges with high-contrast images. The matching block cannot completely represent the details of the image, and the edges of the image will have an edge ringing effect.

2.2. AD image denoising algorithm

The edge information in images increases with the increase in the complexity of the scenes being shot. The BM3D algorithm ignores the characteristics of this edge information. We use anisotropic diffusion to preprocess the image in order to efficiently preserve image details, such as the image texture. The anisotropic diffusion equation is derived from a PDE, which is widely used for noise removal, image edge detection, and detail preservation techniques. Three classical anisotropic diffusion denoising models based on PDEs are discussed below.

Because the thermal diffusion equation smooths the edge and flat regions with the same intensity, which is not conducive to edge retention, P-M proposed a new AD equation [10]. This equation can adaptively change the diffusion coefficient according to the image features, which can help preserve the edge information of an image during denoising. The diffusion model can be expressed as

$$\begin{cases} \dfrac{\partial I(x,y,t)}{\partial t} = div[g(\|\nabla I(x,y,t)\|)\nabla I] \\ I(x,y,0) = I_0(x,y) \end{cases} \tag{1}$$

where $I_0(x,y)$ is the original image and $I(x,y,t)$ is the filtered image after t iterations. $\|\nabla I(x,y,t)\|$ is the gradient modulus after t iterations; it is an edge detector, and its value is smaller in flat areas and larger in edge areas. $g(\cdot)$ is the diffusion coefficient, also known as the "edge stop function", and it represents the degree of diffusion. In particular, when $g(\cdot)$ is constant, the diffusion is isotropic. The diffusion coefficient is a nonnegative monotonically decreasing function that satisfies [24]

$$\begin{cases} \lim_{x \to 0} g(x) = 1 \\ \lim_{x \to \infty} g(x) = 0 \end{cases} \tag{2}$$

In an edge region, where the image gradient varies considerably, the diffusion coefficient $g(\|\nabla I\|)$ can be used to control the image and achieve weaker smoothing, thereby successfully protecting the edge. In a flat region, strong smoothing must be achieved for noise removal.

Perona and Malik constructed the following two classical diffusion coefficients by establishing a relationship between the gradient values and diffusion functions:

$$g_1(\|\nabla I\|) = \frac{1}{1 + (\|\nabla I\|/k)^2} \tag{3}$$

$$g_2(\|\nabla I\|) = \exp[-(\|\nabla I\|/k)^2] \tag{4}$$

where *k* is the diffusion threshold coefficient used to distinguish edges from noise. It is related to the variance of the noise and is used to balance denoising and preserve edges. It can be preset or changed

according to the result of each iteration of the image changes. When $\|\nabla I\| >> k$, $g(\|\nabla I\|) \to 0$, and diffusion is suppressed. When $\|\nabla I\| << k$, $g(\|\nabla I\|) \to 1$, and diffusion is strengthened. Therefore, the selection of an appropriate value for k is crucial for the diffusion behavior of a pixel.

By repeatedly iterating the discrete form of the P-M equation to process the image, noise removal and edge retention can be more appropriately balanced. The discretized PDE shown in Eq (1) can be applied to the image denoising process as follows:

$$I_s^{t+1} = I_s^t + \frac{\lambda}{|\eta_s|} \sum_{p \in \eta_s} g(|\nabla I_{s,p}^t|) \nabla I_{s,p}^t \tag{5}$$

where η_s is the neighborhood space of pixel s, s denotes the coordinates of the pixel, I_s^t is the discrete sample of the current image, λ is a constant that controls the overall diffusion intensity, and $|\eta_s|$ is the size of the neighborhood space. ∇ can be expressed by the difference quotient of two adjacent pixels in different directions:

$$\nabla I_{s,p} = I_t(p) - I_t(s), \ p \in \eta_s = \{N, S, E, W\} \tag{6}$$

For each iteration of the P-M model, the gradient values in the four directions around the center point are calculated, and the gray value of the original center point is replaced by the calculated gradient value, which may induce a loss of image details and false contours.

The P-M equation not only promotes the practical application of nonlinear diffusion equations, but also greatly contributes to the development of PDE methods in the field of image processing. However, the P-M model still has many deficiencies, such as strong noise failure and ill-posedness. Hence, many modifications have been developed to improve the P-M model.

One shortcoming of the P-M model is that it fails when noise is strong. Catte et al. [11] proposed a regularized P-M model called the Catté P-M model. In the Catté P-M model, Gaussian filtering is used to smooth an image and the smoothed gradient mode to replace the gradient mode of the original image. The diffusion coefficient of the P-M AD model is calculated to reduce the noise gradient value. The optimized model can be expressed as

$$\begin{cases} \dfrac{\partial I(x,y,t)}{\partial t} = div[g(\|\nabla I_\sigma\|)\nabla I] \\ I(x,y,0) = I_0(x,y) \end{cases} \tag{7}$$

$$I_\sigma(x,y,t) = G_\sigma * I(x,y,t) \tag{8}$$

where $I(x,y,t)$ is a noisy image, $I_\sigma(x,y,t)$ is a Gaussian smoothed image, G_σ is a Gaussian function with variance of σ^2, and * is a convolutional symbol. The discretized form of the Catté P-M model is

$$I_s^{t+1} = I_s^t + \Delta t \sum_{p \in \eta_s} g(\left|(G_\sigma * \nabla I)_{s,p}^t\right|)\nabla I_{s,p}^t \tag{9}$$

Catté et al. demonstrated that this model is well-posed and ensures the stability of the diffusion process, but the filtering effect will remain poor if the selection of σ is not accurate.

In 2017, Tebini et al. [25] proposed the following diffusion coefficient function, which converges faster and requires less processing time:

$$g_3(\|\nabla I\|) = 1 - \tanh((\|\nabla I\|/k)^2) \tag{10}$$

Tebini et al. discretized the diffusion model as follows:

$$
\begin{aligned}
I(x, y) = I(x, y) + \lambda[&g_N \cdot \nabla_N I + g_S \cdot \nabla_S I + g_E \cdot \nabla_E I + g_W \cdot \nabla_W I + g_{NW} \cdot \nabla_{NW} I \\
&+ g_{NE} \cdot \nabla_{NE} I + g_{SW} \cdot \nabla_{SW} I + g_{SE} \cdot \nabla_{SE} I]_{x,y}
\end{aligned}
\tag{11}
$$

In addition, Tebini et al. demonstrated that the model preserves edges, mitigates staircase effects, and preserves details during the diffusion process.

3. BM3D denoising algorithm based on AD equation

In view of the ringing caused by the Wiener filtering method used in collaborative filtering of the BM3D algorithm, we propose an improved BM3D filtering method based on AD, which combines the advantages of AD and BM3D image denoising algorithms to avoid ringing, retaining clear edges and complete details.
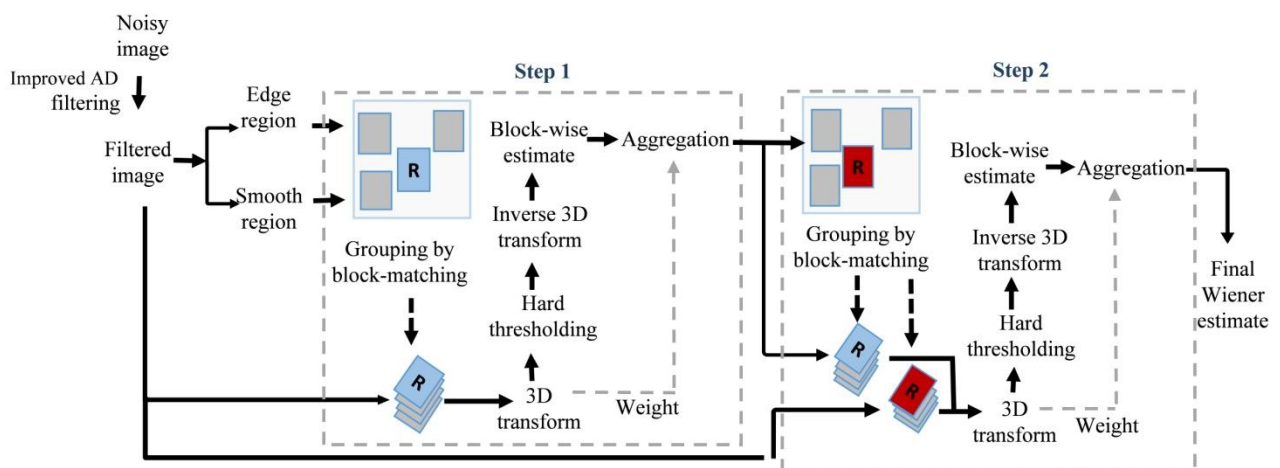
Figure 1. Flow chart of improved algorithm.

The flow of the improved algorithm is shown in Figure 1. First, AD filtering is performed on the noise. Then, the proposed enhancement operator model is utilized to extract the smooth region and edge region of the filtered image. The smooth region is searched horizontally and vertically using the traditional BM3D algorithm. The edge region is searched for similar blocks along the vertical and edge directions [26], and finally denoised images are obtained through grouping, 3D transform,

coefficient shrinking, inverse 3D transform, block estimation, and aggregation. The most significant advantage of the method of searching for similar blocks along the edge direction is that the edge can be obtained for an image with many similar blocks, thus effectively avoiding edge ringing.

The BM3D algorithm is divided into two steps: simple denoising to obtain the basic estimate through the original image matching and more detailed denoising through the original image and basic estimation to improve the PSNR further.

3.1. AD Model based on edge enhancement operator

The direction and diffusion coefficient have considerable effects on the image denoising process. The P-M algorithm was improved in this study in these two aspects. The main procedures are as follows:

- A 5×5 edge enhancement operator template with eight directions was developed to calculate the image gradient. This diffusion direction template can highlight the edge information of the image better than the traditional P-M model and related improved algorithms. The enhanced gradient information is used to calculate the diffusion coefficient, which can enhance the preservation of the image edges and protect details.
- A new diffusion coefficient based on the hyperbolic tangent function was constructed. We prove that the function has an improved convergence speed.

3.1.1. Edge enhancement operator model based on eight-direction diffusion

The classical P-M diffusion model uses the gradient values in four directions in the neighborhood of the pixel to calculate the gray value in the transformation (see Figure 2); it does not consider the effect of the gradient in the diagonal region. Thus, Tebini et al. [25] proposed an improved eight-neighbor diffusion direction model with four additional directions: northeast (NE), northwest (NW), southeast (SE), and southwest (SW) (see Figure 3). The gradient values in the four new directions together with those in the previous four directions are used to calculate the transformed gray value.
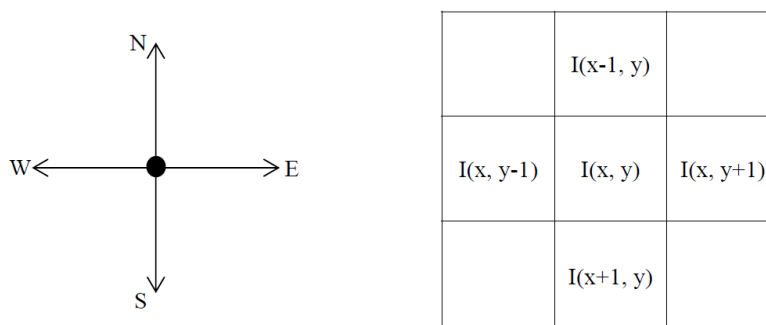
Figure 2. Diffusion directions in the P-M model and relationship with neighboring pixels.

Assuming that the continuous image function is I(x, y), the gradient of $I(x, y)$ at pixel (x, y) is a vector

$$\nabla I(x,y) = \operatorname{grad}(I) = \left(I_x, I_y\right)^{\mathrm{T}} = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)^{\mathrm{T}} \tag{12}$$

In practice, the collected image consists of discrete data in units of pixels, so in digital image processing, the differences between adjacent or interval pixels are often used to represent information at the edge of the image. Therefore, the gradient calculations for the four neighborhood directions of the pixel, *I(x, y)*, can be written as follows:

$$\begin{cases} \nabla_N I_{x,y} = I(x-1, y) - I(x, y) \\ \nabla_S I_{x,y} = I(x+1, y) - I(x, y) \\ \nabla_E I_{x,y} = I(x, y+1) - I(x, y) \\ \nabla_W I_{x,y} = I(x, y-1) - I(x, y) \end{cases} \tag{13}$$

The experimental simulation revealed that although the image edge information detected by the 3×3 template is rich, some image details are still missed. Meanwhile the image edge detected by the 5×5 template is more complete, with clear contour and good continuity. To describe the edge points of the image more accurately and reduce the influence of noise on the detection results, this paper proposes an edge detection template for eight-direction 5×5 enhancement operators. The template is depicted in Figure 4. The distance from a location to the center and the direction for the location in the template determine the weight of each location in the template. Points of equal distance have the same weight.
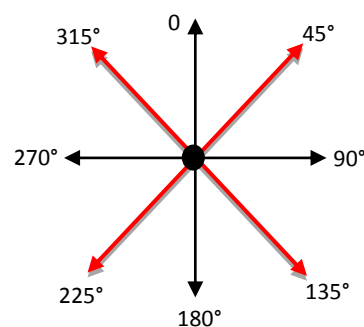
As shown in Figure 4, *w(x, y)* is the weight at each location and can be calculated by using the following equations [27]:

$$d(x,y) = \sqrt{(x-i)^2 + (y-j)^2} \tag{14}$$
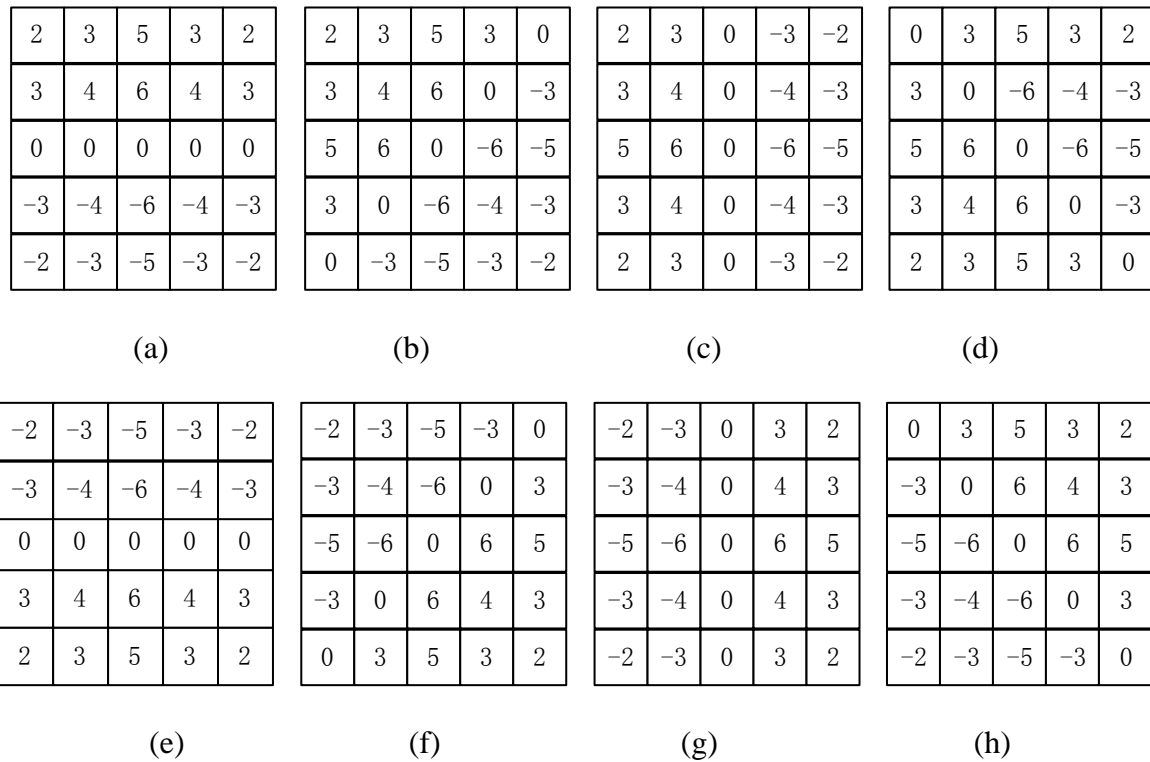
$$\ln s(x,y) = -\ln 2[d(x,y)^2 - u] \tag{15}$$

$$\omega(x,y) = [\mathrm{s}(x,y)] \tag{16}$$

where *d(x, y)* is the Euclidean distance from the template element with coordinates *(x, y)* to the center of the template with coordinates *(i, j)*, u is the adjustment coefficient (related to the template size), and *s(x, y)* is the real weight at *(x, y)*. To simplify the calculations, take the integer at *s(x, y)* to be the element in the template. In the formula, [ ] stands for the up integer operation.

Figure 3. Eight directions of diffusion.

(a)

2	3	5	3	2
3	4	6	4	3
0	0	0	0	0
−3	−4	−6	−4	−3
−2	−3	−5	−3	−2

(b)

2	3	5	3	0
3	4	6	0	−3
5	6	0	−6	−5
3	0	−6	−4	−3
0	−3	−5	−3	−2

(c)

2	3	0	−3	−2
3	4	0	−4	−3
5	6	0	−6	−5
3	4	0	−4	−3
2	3	0	−3	−2

(d)

0	3	5	3	2
3	0	−6	−4	−3
5	6	0	−6	−5
3	4	6	0	−3
2	3	5	3	0

(e)

−2	−3	−5	−3	−2
−3	−4	−6	−4	−3
0	0	0	0	0
3	4	6	4	3
2	3	5	3	2

(f)

−2	−3	−5	−3	0
−3	−4	−6	0	3
−5	−6	0	6	5
−3	0	6	4	3
0	3	5	3	2

(g)

−2	−3	0	3	2
−3	−4	0	4	3
−5	−6	0	6	5
−3	−4	0	4	3
−2	−3	0	3	2

(h)

0	3	5	3	2
−3	0	6	4	3
−5	−6	0	6	5
−3	−4	−6	0	3
−2	−3	−5	−3	0

Figure 4. Enhanced operator templates: (a)-(h) are the 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315° direction detection templates, respectively.

Image I is convoluted with each of the above templates, yielding eight different values for each pixel. The eight values are compared, and the maximum value is assigned to the pixel as the new grayscale value. The convolution process can be expressed as shown in Eq (17)

$$F_{\theta_d}(x, y) = \text{Temp}_{\theta_d} * I_\sigma(x, y) \tag{17}$$

where $I_\sigma(x, y)$ is a Gaussian smoothed image; $d = 1, 2, 3, ..., 8$; and $\theta_1, \theta_2, ..., \theta_8$ are the eight directions of 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315° respectively. Temp_{θ_d} is the detection template corresponding to a direction.

3.1.2. Diffusion coefficient function based on hyperbolic tangent function

The intensity of smoothing in the PDE denoising model is mainly controlled by the diffusion coefficient function. The following diffusion function can be constructed:

$$g_4(\|\nabla I\|) = \exp(-(\|\nabla I\|/k)^2)(1 - \tanh(\|\nabla I\|/k)^2) \tag{18}$$

Compared with the classical diffusion function in the P-M model, this diffusion function has a greater diffusion intensity. The value of $\|\nabla I\|$ is relatively large in the edge area of an image;

therefore, the value of $g(\|\nabla I\|)$ is small. That is, the degree of diffusion is weaker, and the edge of the image can be protected in this case. In a nonedge area, the value of $\|\nabla I\|$ is small; hence, the value of $g(\|\nabla I\|)$ is large, which is favorable for removing noise in flat areas.

Based on the above analysis, the AD model can be discretized as

$$I_s^{t+1} = I_s^t + \lambda \left[\sum_{p \in \eta_s} g\left(\left(\|\nabla I\|\right)_{s,p}^t\right) \nabla I_{s,p}^t + \frac{1}{2} \sum_{p \in \eta_s} g\left(\left(\|\nabla I\|\right)_{s,q}^t\right) \nabla I_{s,q}^t \right] \tag{19}$$

$$\nabla I_{s,p} = I_t(p) - I_t(s), \; p \in \eta_s = \{N, S, E, W\} \tag{20}$$

$$\nabla I_{s,q} = I_t(q) - I_t(s), \; q \in \eta_s = \{NE, SW, SE, NW\} \tag{21}$$

where λ is a constant that controls the overall diffusion intensity. To ensure iteration stability, the value of λ ranges from 0 to 0.25.

Image I is convoluted with each of the above templates, yielding eight different values for each pixel. These eight values are compared, and the maximum value is assigned to the pixel as the new grayscale value. The gray value information is substituted into the diffusion function g, and then g is substituted into the improved discrete AD model to obtain the initial denoising image.

(a)   (b)   (c)   (d)

(e)   (f)   (g)
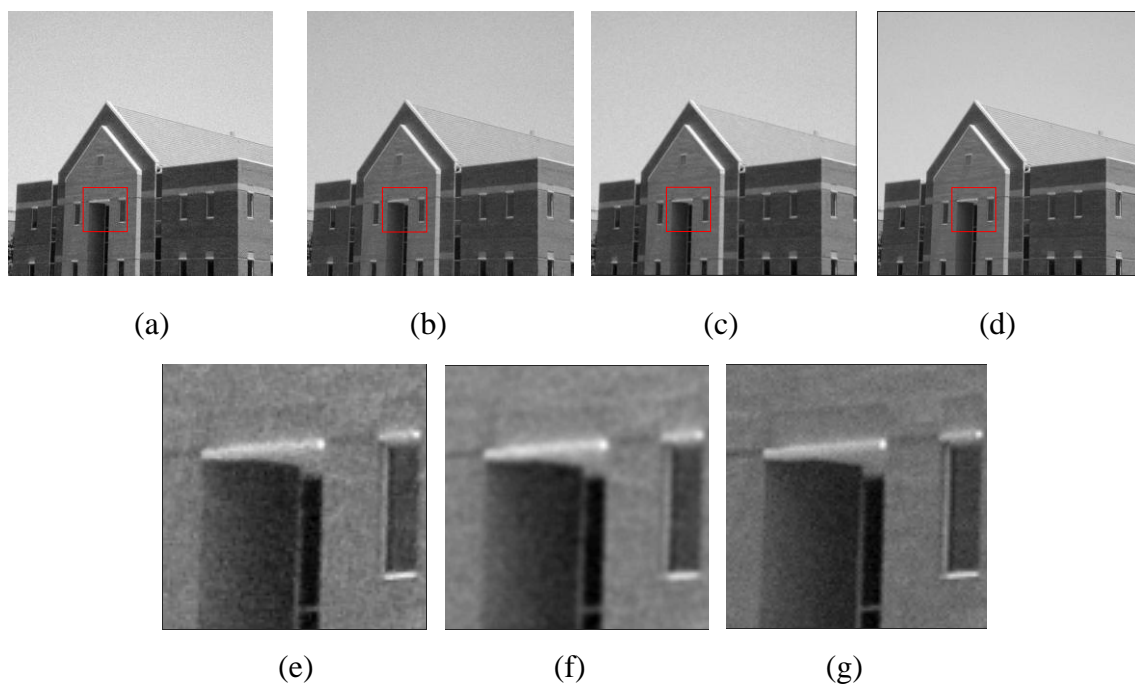
Figure 5. (a) Noise image; images denoised using the (b) P-M method, (c) improved P-M method of Tebini, and (d) proposed method; (e), (f), and (g) enlarged versions of the areas outlined in red in (b), (c), and (d), respectively.

Three methods were used to simulate a house image with Gaussian noise (see Figure 5(a)), in which the noise variance was 10, the gradient threshold k was set to 50, and the number of iterations was set to 5. The experimental results are presented in Figure 5. Figure 5(b) is the denoising image processed using the classical P-M model [8]. Figure 5(c) is the denoising image processed by the improved P-M model of Tebini [23]. Figure 5(d) is the denoising image processed using the method presented in this paper. Figures 5(e), 5(f), and 5(g) are magnified versions of Figures 5(b), 5(c), and 5(d), respectively. It can be seen from Figure 5 that compared with the traditional P-M diffusion model and the improved P-M diffusion model proposed by Tebini, the proposed model retains more image details, more complete image edges, clearer contours, and better continuity. Compared with the P-M model, the denoising effect is obviously improved.

3.2. Mathematical analysis of the proposed model

This section explores the convergence speeds of several diffusion functions. The numerical results and graphical calculations verify the speed and effectiveness of denoising with this model.

Figure 6 displays the decrease in the velocity curve of the proposed function compared to those of the P-M model and other recent functions. We compared and analyzed the output functions corresponding to the different diffusion functions for the same x value. When $x = 40$, the magnitudes

of several diffusion functions correspond to $\begin{cases} g_1(x) = 0.6098 \\ g_2(x) = 0.5273 \\ g_3(x) = 0.4351 \\ g_4(x) = 0.2294 \end{cases}$, indicating that the function value of

the diffusion function $g_4(x)$ is the closest to zero, which demonstrates that $g_4(x)$ converges faster than other models.
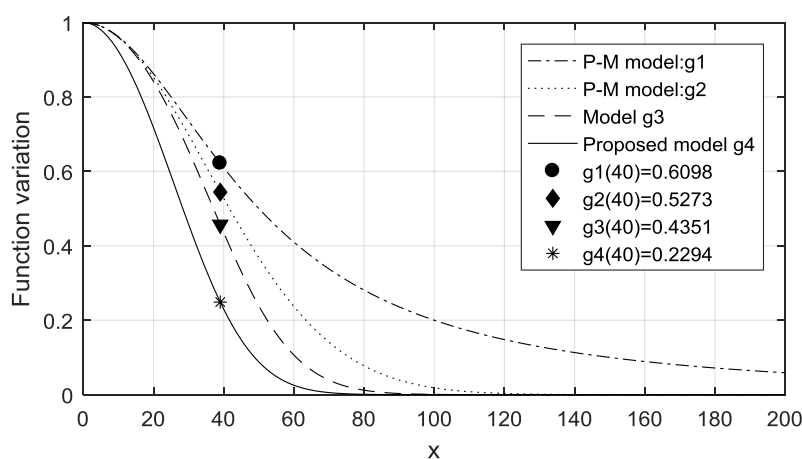
Figure 6. Contrast graphs of the diffusion function curve: function value points of four curves with the same x value.
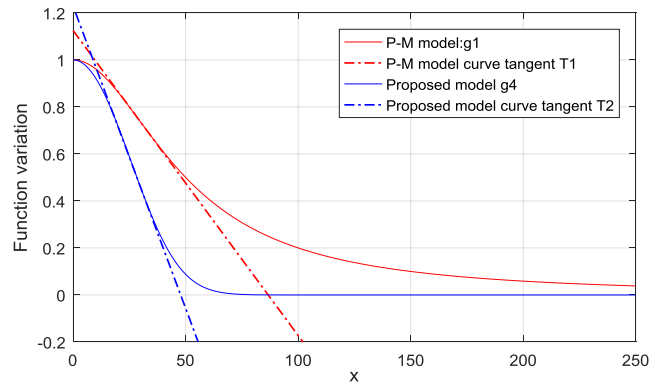
Figure 7. Diffusion function and its tangent.

The trend in a curve can be described by the slope of the tangent line at a point on the curve, namely, the derivative. The slope of the curve reflects the degree of change in the curve at that point. Figure 7 shows the curves for the PM model and the diffusion function of the proposed model, where line segments T1 and T2 are the respective tangent lines. The equation for the tangent line is

$$T = g(z) + g'(z)(x - z) \tag{22}$$

From diffusion function g1 of the P-M model,

$$g_1'(x) = (\frac{1}{1+(x/k)^2})' = -\frac{2x}{k^2} \cdot \frac{1}{(1+(x/k)^2)^2} \tag{23}$$

Thus, the equation for the tangent line of the diffusion function of the P-M model is

$$T_1 = g_1(z) + g_1'(z)(x - z) = \frac{1}{1+(z/k)^2} - \frac{2z}{k^2} \cdot \frac{1}{[1+(z/k)^2]^2} \cdot (x - z) \tag{24}$$

The equation for the tangent line of the proposed model is as follows:

$$T_2 = g_4(z) + g_4'(z)(x - z) \tag{25}$$

According to Eq (18), the slope of the tangent line of the diffusion function of the proposed model can be expressed as follows:

$$g_4'(x) = -\frac{2x}{k^2} \cdot e^{-(x/k)^2} \cdot \left[2 - \tanh(x/k)^2 - \tanh^2(x/k)^2\right] = -\frac{2x}{k^2} \cdot e^{-(x/k)^2} \cdot \left[\frac{6 + e^{-2(x/k)^2}}{(e^{(x/k)^2} + e^{-(x/k)^2})^2}\right] \tag{26}$$

Therefore,

$$g_4'(z) = -\frac{2z}{k^2} \cdot e^{-(z/k)^2} \cdot \left[\frac{6 + e^{-2(z/k)^2}}{(e^{(z/k)^2} + e^{-(z/k)^2})^2}\right] \tag{27}$$

Thus, the equation of the tangent line can be expressed as

$$T_2 = e^{-(z/k)^2}\left[\frac{2e^{-(z/k)^2}}{e^{(z/k)^2} + e^{-(z/k)^2}}\right] - \frac{2z}{k^2}e^{-(z/k)^2}\left[\frac{6 + e^{-2(z/k)^2}}{(e^{(z/k)^2} + e^{-(z/k)^2})^2}\right](x - z) \tag{28}$$

$g_1'(z)$ and $g_4'(z)$ are the slopes of the tangent lines of the diffusion functions of the P-M

and proposed models, respectively. They can be expressed in the form of a difference quotient as follows:

$$\begin{cases} g_1' = \dfrac{dg_1}{dx} = \dfrac{g_1(60) - g_1(40)}{20} = \dfrac{0.4098 - 0.6098}{20} = -0.1 \\ g_4' = \dfrac{dg_4}{dx} = \dfrac{g_4(60) - g_4(40)}{20} = \dfrac{0.02519 - 0.2294}{20} = -0.0102 \end{cases} \Rightarrow \dfrac{dg_4(x)}{dx} > \dfrac{dg_1(x)}{dx} \tag{29}$$

Thus,

$$g_4' > g_1' \tag{30}$$

According to Eq (30), the slope of the diffusion function of the proposed model is larger than that of the P-M model. That is, the variable of the diffusion function g4 changes more rapidly along the curve. Thus, the proposed diffusion function converges faster than that in the P-M model.

4. Experimental results and evaluation

In this section, we experimentally verify the denoising effect of the BM3D filtering model based on the AD equation. To illustrate the performance of the proposed model, images with different noise levels were separately tested. The corresponding results were compared with the results of existing algorithms.

4.1. Image quality evaluation criteria

In the image denoising quality evaluation, we used both objective and subjective evaluation methods to evaluate the effectiveness of the treatment more accurately. Two evaluation criteria for image quality are defined as shown below.

We used the PSNR as an evaluation criterion to measure the approximation of the denoised image to the original clear image. The PSNR can be calculated using

$$PSNR = 10\log_{10}\left[\frac{255 \times 255}{\dfrac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left[u(i,j) - u_0(i,j) \right]^2} \right] \tag{31}$$

where $M \times N$ is the size of the image; $u(i,j)$ and $u_0(i,j)$ are the pixel values of the original and denoised images, respectively, at the corresponding pixel; and L is the range of gray values in the image, where $L = 255$ for 8-bit grayscale images. When the PSNR is larger, the denoised image is closer to the original image.

The second evaluation criterion is the SSIM. The SSIM is an index used to measure the similarity between two images [23]. The SSIM is closer to human visual judgment on image quality and is given by

$$SSIM = \frac{(2\mu_u \mu_{u_0} + c_1)(2\mathrm{cov}_{uu_0} + c_2)}{(\mu_u^2 + \mu_{u_0}^2 + c_1)(\sigma_u^2 + \sigma_{u_0}^2 + c_2)} \tag{32}$$

where μ_u and σ_u^2 are the mean and variance of the image, respectively; cov_{uu_0} is the covariance of u and u_0; and c_1 and c_2 are two very small constants used to prevent the denominator from being zero. A larger SSIM implies higher image similarity.

4.2. Experiments with natural images

To display the visual effect of the denoised image, we firstly added Gaussian white noise with zero mean and variance equal to 20 or 30 to the test images (Figures 8(a) and 9(a), respectively), all images had a size of 512×512. Then we used the proposed method and the existing methods with superior performance, such as BM3D algorithms [5], to denoise the noisy images. Figures 8 and 9 show the corresponding experimental results. The parameters of the proposed anisotropic diffusion method for image preprocessing were set as $\lambda = 1/7$, $k = 50$, and number of iterations were set as 10 for the next four experiments. As can be seen, the improved denoising algorithm provides significantly improved subjective vision compared with the image denoising result of the BM3D algorithm. By enlarging the local information and comparing the results, it can be observed that, by using the proposed algorithm, the face of the photographer and the distant buildings in Figure 8(h) are clearer than those in Figure 8(g). Ringing is visible at the edges of Figure 9(i), while ringing is effectively avoided in Figure 9(j). Furthermore, as shown in Figure 9(h), information such as the ring and bracelet on the hand of the man are effectively saved. The algorithm is more efficient at retaining the image texture.
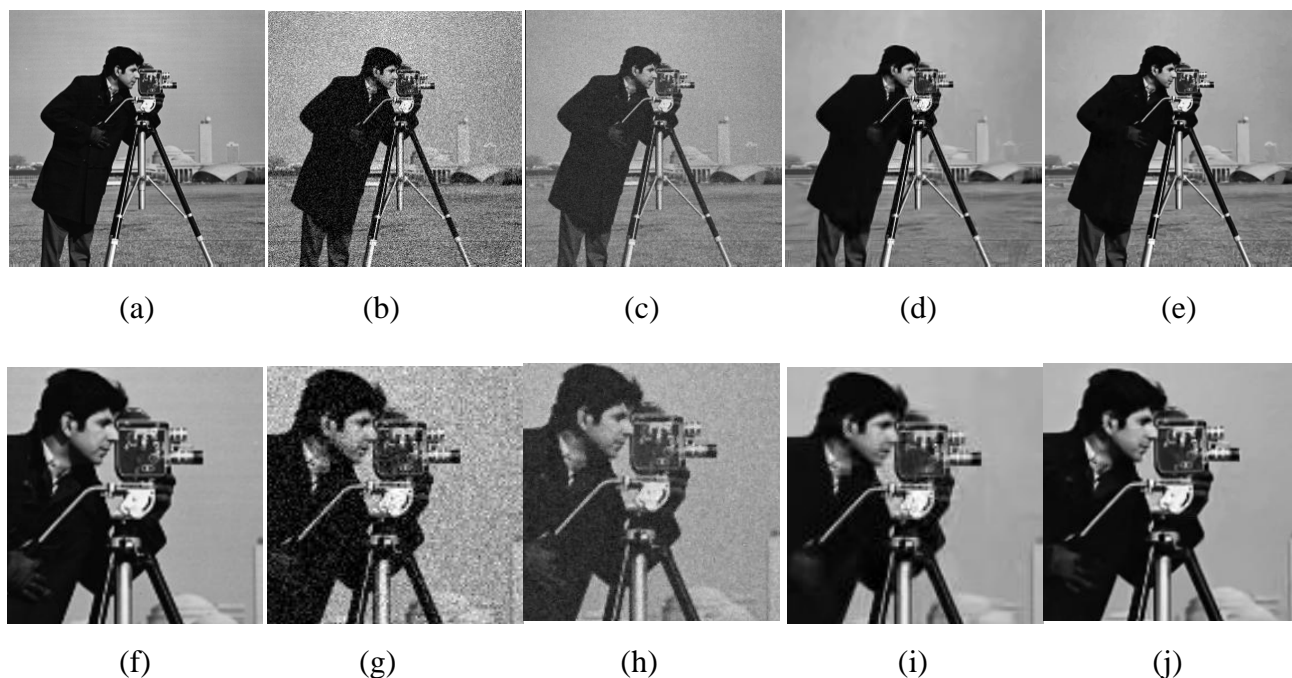
Figure 8. (a) Original image; (b) noise image with variance = 20; image after denoising using the (c) P-M method, (d) BM3D method, and (e) proposed method; (f), (g), (h), (i), and (j) enlarged portions of (a), (b), (c), (d), and (e), respectively.
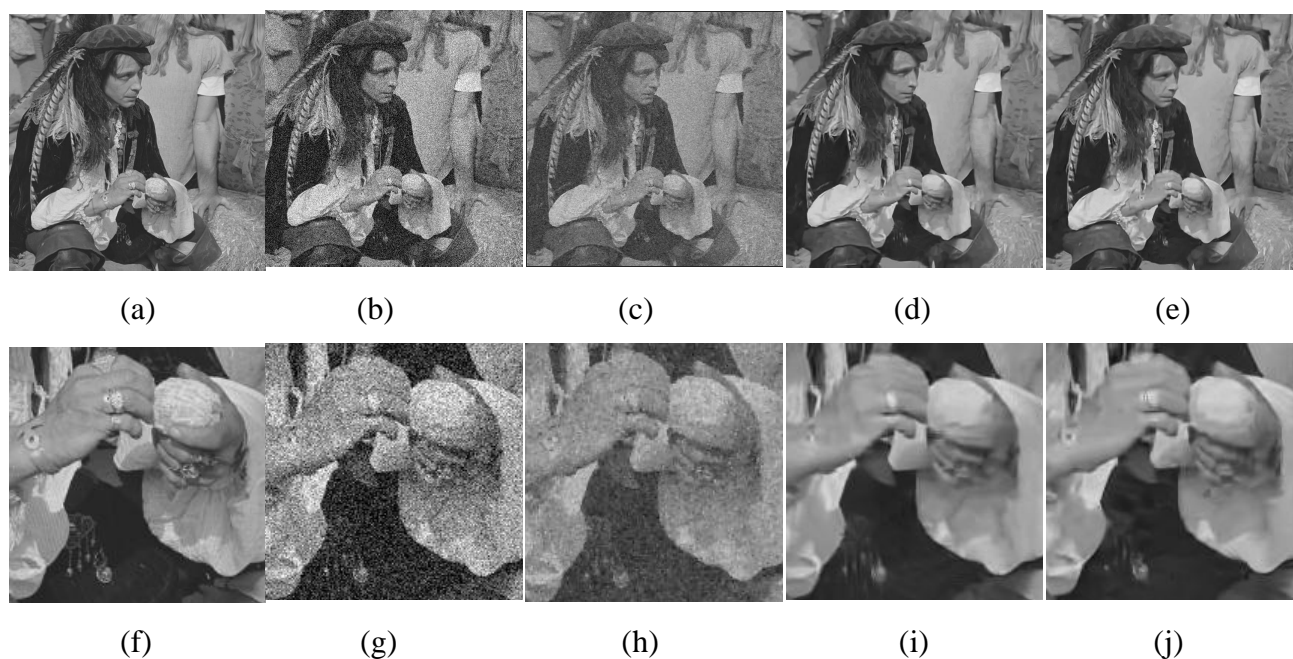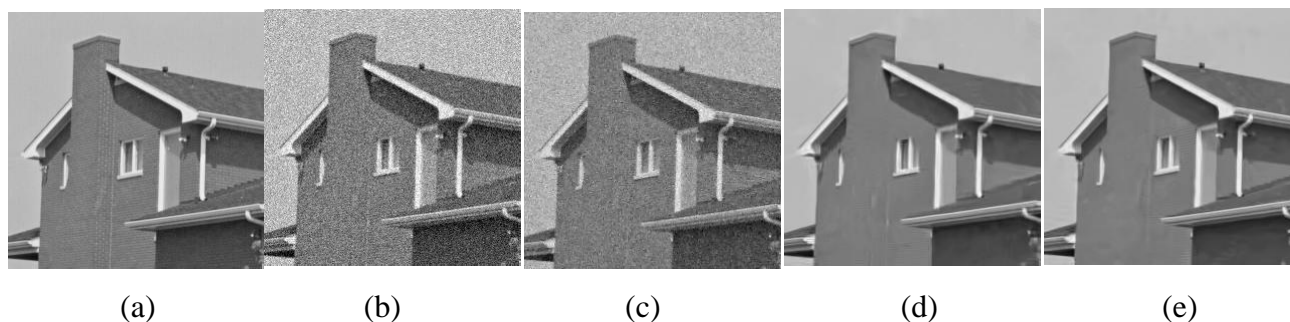
Figure 9. (a) Original image; (b) noise image with variance = 30; image after denoising using the (c) P-M method, (d) BM3D method, and (e) proposed method; (f), (g), (h), (i), and (j) enlarged portions of (a), (b), (c), (d), and (e), respectively.

To verify the generality of the proposed algorithm for image denoising, two more images were selected for simulation experiments. Gaussian white noise with zero mean and variance equal to 20 or 35 was added to the images of a house (512×512) and CT (512×512) (Figures 10(a) and 11(a), respectively). The P-M denoising method, BM3D denoising method, and proposed method were used to denoise the test image after adding noise. The experimental results are shown in Figures 10 and 11. Comparative analysis revealed that the proposed algorithm retains the texture details of the images more effectively.

To confirm the denoising performance of the proposed algorithm, different noises with zero mean and variance equal to 10 were selected to perform noise processing on the four abovementioned test images. The relevant parameters are set as $\lambda = 0.1$, $k = 50$, and number of iterations were set as 5. Experiments were performed on the noise-enhanced images using the non-local mean, BM3D, and proposed denoising algorithms. The PSNR and SSIM were utilized to measure the image denoising effect, and the results are presented in Table 1.

The time required for the noise image processing by three different algorithms is shown in Table 2. To ensure accuracy, each time value measurement is performed by repeated denoising for the same noise image 50 times, and their average value is taken. It can be seen from Table 1 that compared with the results of the non-local mean algorithm and the three-dimensional block matching filtering algorithm, the proposed algorithm has a higher SSIM and PSNR. As can be seen from Table 2, the time required by the proposed algorithm to process noisy images is at most 0.2 seconds longer than the original BM3D algorithm.

Figure 10. (a) Original image; (b) noise image with variance = 20; image after denoising using the (c) P-M method, (d) BM3D method, and (e) proposed method.
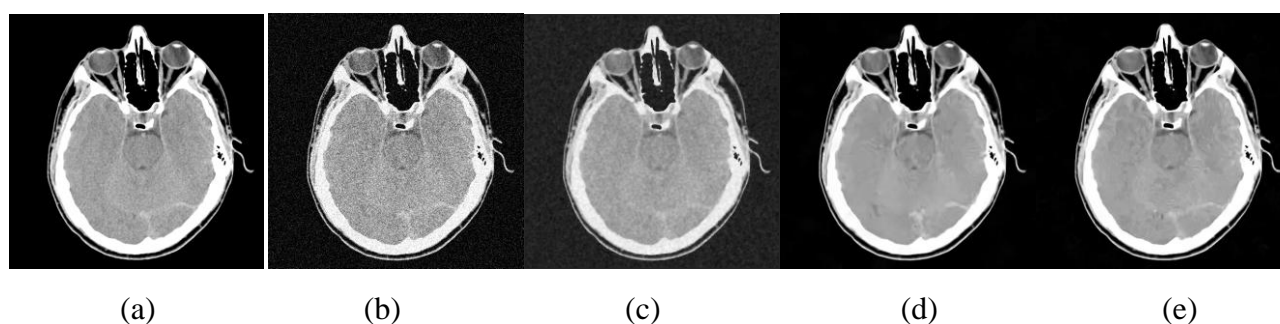
Figure 11. (a) Original image; (b) noise image with variance = 35; image after denoising the (c) P-M method, (d) BM3D method, and (e) proposed method.

**Table 1.** Comparison of denoising results in terms of PSNR (dB)/SSIM.

Image	$\sigma / PSNR$	NLM		BM3D		Our method	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Cameraman	10	29.42	0.877	34.18	0.929	37.63	0.969
(256×256)	20	28.50	0.824	30.48	0.871	34.65	0.945
	30	26.95	0.681	28.64	0.832	32.11	0.916
	40	24.99	0.534	27.93	0.801	30.57	0.891
House	10	34.51	0.883	36.36	0.908	36.94	0.901
(256×256)	20	32.20	0.827	33.45	0.869	34.71	0.884
	30	29.27	0.816	32.09	0.847	32.65	0.854
	40	26.52	0.784	30.75	0.827	31.10	0.833
Man	10	31.29	0.921	33.32	0.957	34.79	0.970
(512×512)	20	29.41	0.879	30.43	0.912	31.11	0.926
	30	27.48	0.831	28.81	0.865	29.28	0.885
	40	25.35	0.765	27.62	0.832	27.97	0.844
CT	10	41.85	0.938	45.23	0.961	46.06	0.975
(512×512)	20	38.51	0.907	41.11	0.931	42.07	0.949
	30	29.99	0.526	38.14	0.897	39.47	0.914
	40	26.25	0.349	37.20	0.875	37.77	0.907

Table 2. Comparison of running time of different denoising algorithms.

Image	σ	Time(s)		
		NLM	**BM3D**	**Our method**
Cameraman	10	64.31	0.98	1.08
(256×256)	20	65.63	1.09	1.14
	30	66.73	1.12	1.20
	40	68.35	1.09	1.20
House	10	71.56	1.03	1.07
(256×256)	20	71.86	1.09	1.16
	30	72.69	1.14	1.20
	40	74.07	1.12	1.24
Man	10	262.28	2.81	2.95
(512×512)	20	263.89	2.95	3.15
	30	264.17	3.21	3.27
	40	266.03	3.10	3.17
CT	10	267.86	3.15	3.21
(512×512)	20	267.90	3.17	3.29
	30	268.42	3.20	3.25
	40	269.87	3.12	3.21

To reflect the superiority of the new algorithm more intuitively, we set the abovementioned cameraman, man, and house images to a mean of 0 and noise deviation in increments of 10, using the BM3D filtering and proposed algorithms. The changes in the PSNR and SSIM with application of the two denoising models are depicted as the noise level changes, as shown in Figures 13 and 14, respectively. The triangles represent the BM3D denoising algorithm, and the dots represent the proposed algorithm. This figure shows intuitively that the proposed method has a better denoising effect.
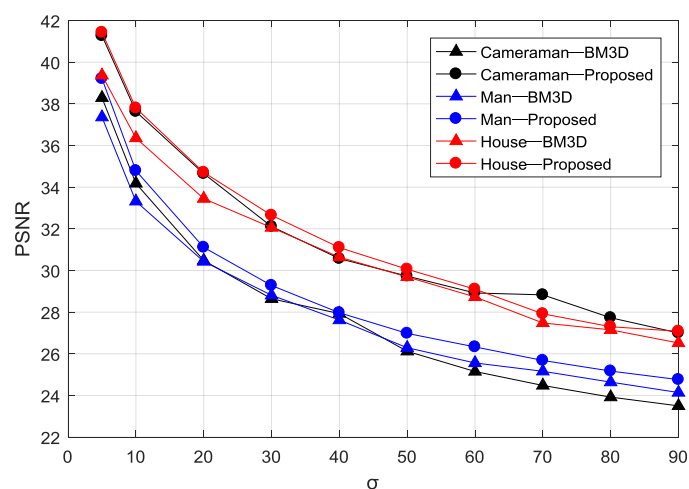
Figure 13. PSNR after denoising using different algorithms.
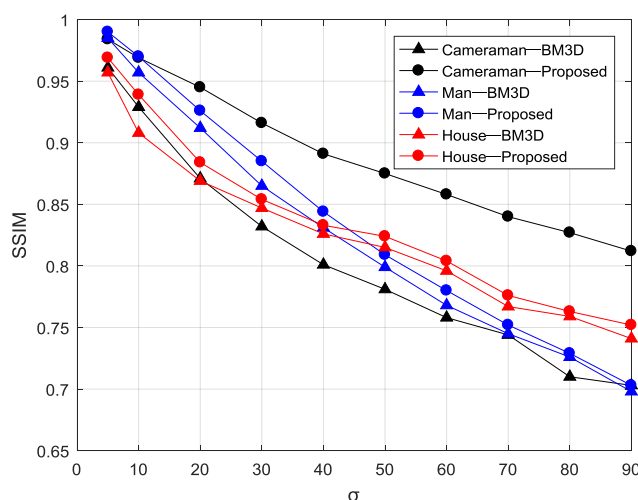
Figure 14. SSIM after denoising using different algorithms.

5. Conclusion

This paper proposed an improved BM3D denoising algorithm. In this approach, AD filtering is firstly performed on the noisy image; then, similar blocks are searched for in the vertical and edge directions to implement denoising. Finally, the denoised image is obtained. This paper also presented a new AD model for noise reduction and edge preservation. The restored image was obtained for visual evaluation. It is evident that the improved method avoids the edge ringing produced by the traditional denoising methods. In addition, the values of the two objective evaluation criteria—the PSNR and SSIM—showed that the proposed model performs better than BM3D methods, demonstrating that the proposed method is superior in terms of noise removal and edge and detail preservation.

However, the improved method also increases the time complexity while improving the denoising performance. Therefore, investigations to improve the denoising performance and denoising efficiency will be the focuses of future research. In addition, the model was evaluated using images with Gaussian noise, and images with other types of noise were not used. Furthermore, the theory only focuses on images. Therefore, in the future, we plan to address the time complexity, improve the denoising performance of the algorithm further, and explore the application of AD to video processing.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. U. Erkan, D. N. H. Thanh, L. M. Hieu, S. Engínoğlu, An Iterative Mean Filter for Image Denoising, *IEEE Access*, **7** (2019), 167847–167859.

2. Z. Wang, X. Tan, Q. Yu, J. Zhu, Sparse PDE for SAR image speckle suppression, *IET Image Process.,* **11** (2017), 425–432.

3. Y. Wu, G. Gao, C. Cui, Improved Wavelet Denoising by Non-Convex Sparse Regularization Under Double Wavelet Domains, *IEEE Access*, **7** (2019), 30659–30671.

4. A. Buades, B. Coll, J. M. Morel, A non-local algorithm for image denoising, *2005 IEEE Comput. Soc. Conference Comput. Vis. Pattern Recognit.*, **2** (2005), 60–65.

5. K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering, *IEEE Trans. Image Process.*, **16** (2007), 2080–2095.

6. H. Zhong, K. Ma, Y. Zhou, Modified BM3D algorithm for image denoising using nonlocal centralization prior, *Signal Process.*, **106** (2015), 342–347.

7. B. Shi, Q. Lian, S. Chen, X. Fan, SBM3D: Sparse Regularization Model Induced by BM3D for Weighted Diffraction Imaging, *IEEE Access*, **6** (2018), 46266–46280.

8. G. Chen, G. Luo, L. Tian, A. Chen, Noise Reduction for Images with Non-uniform Noise Using Adaptive Block Matching 3D Filtering, *Chinese J. Electron.*, **26** (2017), 1227–1232.

9. Y. Li, J. Zhang, M. Wang, Improved BM3D denoising method, *IET Image Processing*, **11** (2017), 1197–1204.

10. P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Anal. Mach. Intell.,* **12** (1990), 629–639.

11. F. Catté, P. L. Lions, J. M. Moresl, T. Coll, Image Selective Smoothing and Edge Detection by Nonlinear Diffusion, *SIAM J. Numer. Anal.*, **29** (1992), 845–866.

12. G. Gilboa, N. Sochen, Y. Y. Zeevi, Image enhancement and denoising by complex diffusion processes, *IEEE Trans. Pattern Anal. Mach. Intell.*, **26** (2004), 1020–1036.

13. M. J. Black, G. Sapiro, D. H. Marimont, D. Heeger, Robust anisotropic diffusion, *IEEE Trans. Image Process.*, **7** (1998), 421–432.

14. V. Bhateja, G. Singh, A. Srivastava, J. Singh, Speckle reduction in ultrasound images using an improved conductance function based on Anisotropic Diffusion, *2014 Int. Conference on Comput. Sustain. Glob. Dev.,* IEEE, (2014), 619–624.

15. C. Tsiotsios, M. Petrou, On the choice of the parameters for anisotropic diffusion in image processing, *Pattern Recognit.*, **46** (2013), 1369–1381.

16. S. Tebini, Z. Mbarki, H. Seddik, E. B. Breik, Rapid and efficient image restoration technique based on new adaptive anisotropic diffusion function, *Digit. Signal Process.*, **48** (2016), 201–215.

17. T. F. Chan, S. Esedoglu, F. Park, A fourth order dual method for staircase reduction in texture extraction and image restoration problems, *2010 IEEE Int. Conference Image Process.*, Hong Kong, (2010), 4137–4140.

18. G. Motta, E. Ordentlich, I. Ramirez, G. Seroussi, M. J. Weinberger, The iDUDE Framework for Grayscale Image Denoising, *IEEE Trans. Image Process.*, **20** (2010), 1–21.

19. K. Liu, J. Tan, B. Su, Adaptive Anisotropic Diffusion for Image Denoising Based on Structure Tensor, *2014 5th Int. Conference Digit. Home*, IEEE, (2014), 111–116.

20. Y. Q. Wang, J. Guo, W. Chen, W. Zhang, Image denoising using modified Perona–Malik model based on directional Laplacian, *Signal Process.*, **93** (2013), 2548–2558.

21. H. Yu, C. S. Chua, GVF-based anisotropic diffusion models, *IEEE Trans. Image Process.*, **15** (2006), 1517–1524.

22. H. Tian, H. Cai, J. H. Lai, X. Xu, Effective image noise removal based on difference eigenvalue, *18th IEEE Int. Conference Image Process.,* (2011), 3357–3360.

23. Y. Toufique, R. C. E. Moursli, L. Masmoudi, A. E. Kharrim, M. Kaci, S. Allal, Ultrasound image enhancement using an adaptive anisotropic diffusion filter, *2nd Middle East Conference Biomed. Eng.*, IEEE, (2014), 1–4.

24. H. S. Kim, J. M. Yoo, M. S. Park, T. N. Dinh, G. S. Lee, An Anisotropic Diffusion Based on Diagonal Edges, *IEEE 9th Int. Conference Adv. Commun. Technol.,* (2007), 384–388.

25. Z. Mbarki, H. Seddik, S. Tebini, E. B. Braiek, A new rapid auto-adapting diffusion function for adaptive anisotropic image de-noising and sharply conserved edges, *Comput. Math. Appl.,* **74** (2017), 1751–1768.

26. J. Liu, R. Liu, Y. Wang, J. Chen, Y. Yang, D. Mag, Image denoising searching similar blocks along edge directions, *Signal Process. Image Commun.,* **57** (2017), 33–45.

27. Y. Zhang, X. Han, H. Zhang, L. Zhao, Edge detection algorithm of image fusion based on improved Sobel operator, *IEEE 3rd Inform. Technol. Mechatron. Eng. Conference (ITOEC)*, (2017), 457–461.