

## Research Article

# Time Parallel Denoising Algorithm Based on P-M Equation for Real Image

Guoliang He 

*School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China*

Correspondence should be addressed to Guoliang He; hegl@uestc.edu.cn

Received 29 June 2022; Accepted 15 July 2022; Published 11 August 2022

Academic Editor: Chia-Huei Wu

Copyright © 2022 Guoliang He. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper combines the regularized Perona and Malik model with the time parallel algorithm to denoise the real image which has the characteristics of typical Gaussian noise. This method can remove the noise stably and effectively and has high calculation efficiency. For the real color image of  $6000 \times 4000$  pixels, the time cost can be shortened from one minute in serial computing to a few seconds based on our combination method if we do not care about the requirement of computing hardware. This provides some reference for practical engineering application.

## 1. Introduction

Image denoising is a process of reducing noise in digital image. Digital image is often contaminated by noise from imaging equipment and external environment in the process of digitization and transmission, which makes an image have a variety of noises in practical applications. These noises cause a lot of detrimental effects on the analysis and application of image. In addition, the process of image regeneration will be inevitably be polluted by noise, which also damages the quality of the image. These noises not only influence people's visual effect but also lead to some difficulty for the subsequent analysis of the image. Therefore, it is necessary to denoise the image in order to improve signal to noise ratio (SNR).

Traditional image denoising methods include the following [1–3]: mean filter (MF), geometric mean filter (GMF), inverse harmonic mean filter (IHMF), Wiener filter (WF), median filter (MF), morphological noise filter (MNF), wavelet denoising (WD), variational filtering (VF), and differential equation filtering (DEF). The MF uses the neighborhood average method, which is suitable for removing the particle noise in the scanned image. GMF can

achieve the same smoothness as MF, but more image details will be lost in the process of filtering. The IHMF is suitable for dealing with impulse noise and has good effect on “salt” noise, but it needs to know in advance whether the noise is dark noise or bright noise; in other words, selecting an appropriate filter order is necessary. If the order symbol selected is inappropriate, it may cause very bad consequences. AWF can adjust the output of the noisy image according to the local variance of the image. The larger the local variance is, the stronger the smoothing effect of the filter is. The filtering effect of this method is better than that of the MF, and it is useful to retain the edge of the image and other high-frequency parts, but the calculation is also large. MF is a commonly used nonlinear smoothing filter. Its basic idea is to replace the value of a point in a digital image or digital sequence with the median value of each point in a field of that point, so that isolated noise points can be eliminated. Therefore, MF is very effective for filtering salt and pepper noise in an image. However, if an image has many details, especially for those with many points, lines, and spires, median filtering is not suitable for image noise reduction. Morphological noise filter combines open operation and close operation to filter noise. The applicable image type

of this method is that the object size in the image is relatively large, and there is no small details.

General speaking, according to the characteristics of high frequency noise, neighborhood average method can effectively suppress the noise, but it can also cause the fuzzy phenomenon. The fuzzy degree is usually proportional to the domain radius [2, 4, 5].

In addition, wavelet denoising, total variation denoising model, such as TV model, deep learning-based denoising methods, such as NL means method, etc., are good at dealing with some problems at present [6–9], but the noise processing of real image with large size is still of inefficiency. They all inevitably face the same problem: although the CPU's or GPU's performance has been improved by hundreds of times, it is far from meeting the real-time computing needs, which largely limits the use of these algorithms because people cannot bear to wait for several minutes to process a single photograph. Therefore, there is still a distance between these denoising methods and the practical application.

The real image noise is usually signal related, but there is no clear distribution to simulate, which brings great difficulties to the practical application. Generally speaking, the global distribution of noise image is not Gaussian distribution, but the noise of different channels of all local image blocks can be approximately regarded as the Gaussian distribution with different variance [11, 12], which provides a good theoretical support for deducing the actual image noise.

PDE method, a kind of image processing method which rises in recent years, has achieved good results in image processing [13–15]. This kind of method has the characteristics of anisotropy. In image denoising, it can depress the noise and keep the edge well, especially for Gaussian type noise. Recently, the research of image processing method based on PDE is a hot research direction of image denoising and has achieved some theoretical and practical application results. The basic idea of denoising method based on differential equation is to establish an initial condition of noise image as a certain nonlinear PDE (i.e., filtering results), using the diffusion characteristics of differential equation to continuously reduce the high-frequency part, solving the diffusion PDE, and getting the solution at different times. The solution of PDE mainly adopts the time iteration method, which makes the image approach to the desired mode gradually through the iteration. The typical representative of this algorithm is Perona and Malik's equation as well as the subsequent improvement [16]. This method has a large selection in determining the diffusion coefficient and has the function of backward diffusion as well as forward diffusion. Therefore, it has the ability of smoothing the image and sharpening the edge. Besides, a large number of experiments show that this kind of method has a good effect in removing Gaussian noise [11, 12].

It is also worth noting that although PDE method as mentioned above has achieved good results in image processing with low noise density, the denoising effect is not good in image processing with high noise variance, and the processing time is significantly higher than low density. Time parallel-based algorithm is a fast algorithm to solve

time-related problems, which can effectively improve the efficiency of calculation [17–19]. It is a desirable method in the situation of high demand for calculation time. It should be noted that although large-scale parallel computing devices do not seem to be available frequently, with the rapid development of cloud computing and edge computing, the computing environment supported by hardware will be greatly improved, thus providing more possibilities and opportunities for image parallel computing [10].

Perona and Malik proposed a PDE filter-based nonlinear diffusion model (P-M model) [16]. P-M is a nonlinear anisotropic method, which aims to overcome the shortcomings of fuzzy edge and edge position moving in linear filtering method. The basic idea is to reduce the diffusion coefficient where the image features are strong and enhance the diffusion coefficient where the image features are weak. The anisotropic denoising model determines the diffusion speed according to the gradient value of the image, so that it can meet the requirements of both noise depression and edge preservation. These models, represented by P-M equation, have been widely used in image enhancement, image segmentation, edge detection, and other fields and achieved good results. The below part gives a brief introduction to the P-M model.

Perona and Malik first proposed a nonlinear P-M model that can maintain the boundary. The primary P-M model is:

$$\begin{cases} \frac{\partial I(x, y, t)}{\partial t} = \text{div} ((g(|\nabla I|) \cdot \nabla I), \\ I(x, y, 0) = I_0, \end{cases} \quad (1)$$

where  $I_0$  is the original image,  $I(x, y, t)$  is the diffusion image of the original image at time  $t$ ,  $\nabla$  is the gradient operator,  $\text{div}$  is the  $g(r) = 1$  divergence operator, and  $g(r)$  is the diffusion coefficient. The ideal diffusion coefficient  $g(r)$  should make the anisotropic diffusion fast in the region where the gray level changes gently and low-speed diffusion or even non-proliferation function in the position where the gray level changes sharply (i.e., the image feature), so  $g(r) > 0$  is a non-increasing function. If the model degenerates into heat conduction equation, Perona and Malik give two diffusion coefficients:

$$g(r) = e^{-(r/K)^2}, \quad (2)$$

where the constant  $K$  is related to the variance of noise. P-M model is an improved partial differential equation of heat conduction, which can adaptively control the diffusion velocity by the function  $g(r)$ . The P-M model makes use of the function  $g(r)$  to control the diffusion speed, that is to say, the method increases the smoothing intensity inside the region while decreases the smoothing intensity at the edge. In the P-M model, the diffusion size is determined by the absolute value of the gradient of the pixel. If the absolute value of the gradient of a point in the image is greater than the  $K$  value, the gradient value will slowly decrease with the increase of the number of iterations in the process of iteration, playing the role of maintaining the edge. However,

if the absolute value of the gradient of the point is less than the  $K$  value, the gradient value will soon tend to 0 in the process of iteration, playing a smooth effect. In the P-M model, the edge is identified by the gradient differential operator.

It should be noted that P-M Equation (1) can bring good results in suppressing noise and preserving important features of image, but it is ill-conditioned and unstable. In order to eliminate the “*ill-condition*” of the P-M equation and maintain the well posed property, the image is presmoothed by using the original Gaussian convoluting image. Catta introduces a regular P-M model in [20]:

$$I_t = \text{div} (g(|\nabla I_\sigma|)\nabla I), \quad (3)$$

where  $I_\sigma(x, y, t) = G_\sigma * I(x, y, t)$ . Here,  $G_\sigma$  is the Gaussian function with variance  $\sigma$ , and  $*$  is convolution. The regularized P-M model can eliminate the noise better than the mean value method and can overcome the ill-pose of the initial value of the P-M model. This model (3) is a regularization model in space domain, which overcomes the sensitive  $\nabla I$  relation to noise. We can prove that the solution of the regularization model is unique, and the solution is stably depended on initial value continuously [20, 21]. This paper is aimed at developing the numerical solution of the second order nonlinear diffusion Equation (3) efficiently.

## 2. Numerical Method of the P-M Model

For the P-M equation, the classical finite difference time-space total discretization mainly has two kinds of schemes: explicit and implicit. The explicit method does not need to solve large discrete linear equations and can directly compute large images. However, this method is very strict to the time step and the space step, which results in that the image dimension discretized directly can be very large and costs much iteration time actually. On the other hand, the implicit method has no limitation on the time discretization, but it needs to solve a set of equations repeatedly. If the implicit method is used to do with the large image, because the dimension of discrete equations is very huge and the singularity is very serious, it inevitably brings the low computational efficiency.

Aiming at P-M model, this paper combines explicit method, implicit method, and time parareal algorithm to construct a new time parallel iterative format. This algorithm adopts explicit computing in relatively fine time grids and implicit computing in rough time grids and accelerates the iteration by time parareal method. It can deal with large image directly and save more time in denoising.

For ease of discretization, the two-dimensional regularized P-M equation can be rewritten in the following form (for detailed introduction, please refer to literature [16, 20]):

$$\partial I_t = \partial_x [g(|\nabla I_\sigma|) \cdot \partial I_x] + \partial_y [g(|\nabla I_\sigma|) \cdot \partial I_y]. \quad (4)$$

Let  $N_{\max x}$ ,  $N_{\max y}$ , and  $N_{\max t}$  represent the maximum iteration steps in the  $X$  direction,  $Y$  direction, and time direction, respectively. For the sake of simplicity, we take the equal steps in the  $x$  direction and  $y$  direction, that is,  $\Delta$

$x = \Delta y = h$ , and  $\Delta t = \tau$  is the time step, the grid coordinate is  $(x_i, y_j, t_n) = (ih, jh, n\tau)$ ,  $0 < i \leq N_{\max x}$ ,  $0 < j \leq N_{\max y}$ ,  $0 < n \leq N_t = N_{\max t}$ ,  $I_{i,j}^n$  is the approximate value of  $I(ih, jh, n\tau)$ , and let  $g_{i,j} = g(|\nabla I_\sigma|)_{i,j}$ .

The time explicit difference format is introduced first. In reference [13], Perona and Malik give the following explicit difference scheme to solve Equation (4):

$$I_{i,j}^{n+1} = I_{i,j}^n + \lambda \cdot (g_N^n \cdot \nabla_N^n I + g_S^n \cdot \nabla_S^n I + g_E^n \cdot \nabla_E^n I + g_W^n \cdot \nabla_W^n I). \quad (5)$$

In the above explicit scheme (5),  $N$ ,  $S$ ,  $E$ , and  $W$  present North, South, East, and West, respectively, and  $\lambda = \tau/h$ . Because the region normalized  $h$  is generally small, for example,  $h = 0.001$ , the stability of numerical solution for the explicit relation (5) needs theoretically the time step is  $\lambda = 1000\tau$ ,  $\lambda < 0.25$ , namely,  $\tau < 0.00025$  is required. If the symbol  $\nabla$  in formula (5), difference quotient, is used, the specific discrete form is follows:

$$\begin{aligned} \nabla_N I_{i,j} &= I_{i,j+1} - I_{i,j}, \quad \nabla_E I_{i,j} = I_{i+1,j} - I_{i,j}, \\ \nabla_S I_{i,j} &= I_{i,j-1} - I_{i,j}, \quad \nabla_W I_{i,j} = I_{i-1,j} - I_{i,j}. \end{aligned} \quad (6)$$

Let us see the time semi-implicit difference scheme. The right side of Equation (3) can be written as:

$$\text{div} (g\nabla I) = \frac{\partial}{\partial x} \left( g \frac{\partial I}{\partial x} \right) + \frac{\partial}{\partial y} \left( g \frac{\partial I}{\partial y} \right). \quad (7)$$

In practical application, a divergence discretization format called “*half point*” discretization [13, 23] is often used. The main idea includes the following steps.

First, calculate the value of  $gI_x$  at “*half point*” in the “East, West” direction and “*half point*” in the “South, North” direction, respectively, and then, the implicit format is obtained as follows:

$$\begin{aligned} I_{i,j}^{n+1} &= I_{i,j}^n + \tau \cdot \left( \frac{g_{i-1,j}^n + g_{i,j}^n}{2} \cdot (I_{i-1,j}^{n+1} - I_{i,j}^{n+1}) + \frac{g_{i+1,j}^n + g_{i,j}^n}{2} \cdot (I_{i+1,j}^{n+1} - I_{i,j}^{n+1}) \right. \\ &\quad \left. + \frac{g_{i,j-1}^n + g_{i,j}^n}{2} \cdot (I_{i,j-1}^{n+1} - I_{i,j}^{n+1}) + \frac{g_{i,j+1}^n + g_{i,j}^n}{2} \cdot (I_{i,j+1}^{n+1} - I_{i,j}^{n+1}) \right). \end{aligned} \quad (8)$$

In the above formula (8), the value of  $g(r)$  at “*half point*” is approximated by the average of two adjacent whole points, that is,

$$\begin{aligned} g_{i,j+(1/2)} &\approx \frac{(g_{i,j+1} + g_{i,j})}{2, \dots, g_{i,j-(1/2)}} \approx \frac{(g_{i,j-1} + g_{i,j})}{2}, \\ g_{i+(1/2),j} &\approx \frac{(g_{i+1,j} + g_{i,j})}{2, \dots, g_{i-(1/2),j}} \approx \frac{(g_{i-1,j} + g_{i,j})}{2}. \end{aligned} \quad (9)$$

Thus, by using semi-implicit scheme, a system of linear equations is obtained, which is represented by matrix as

follows:

$$\mathbf{I}^{n+1} = \mathbf{I}^n + \tau \cdot \mathbf{A}^n \cdot \mathbf{I}_n^{n+1}, \quad (10)$$

where  $\mathbf{I}_{ij}^n$  and  $\mathbf{I}_{ij}^{n+1}$  present the approximate image for the  $n$ th time step and next time step, respectively. Equation (10) is an implicit format, and  $\tau$  is a time step. The above formula can be written as follows:

$$\mathbf{I}^{n+1} = (1 - \tau \cdot \mathbf{A}^n)^{-1} \mathbf{I}^n. \quad (11)$$

If  $\mathbf{I}^{n+1}$  or  $\mathbf{I}^n$  is a matrix of  $M \times N$ , then matrix  $\mathbf{A}$  is a dimensional sparse matrix, but not a tridiagonal matrix. Because the coefficient matrix is very large and the solution efficiency is not high, Weickert and Scharr proposed the additive operator splitting (AOS) algorithm in [22]:

$$\begin{cases} (1 - 2\tau \cdot \mathbf{A}_x^n) \mathbf{I}_1^{n+1} = \mathbf{I}^n, \\ (1 - 2\tau \cdot \mathbf{A}_y^n) \mathbf{I}_2^{n+1} = \mathbf{I}^n. \end{cases} \quad (12)$$

Then, the mean value of the two terms is calculated as a complete iterative process:

$$\mathbf{I}^{n+1} = \frac{1}{2} (\mathbf{I}_1^{n+1} + \mathbf{I}_2^{n+1}). \quad (13)$$

Note: it is worthy of note that AOS algorithm is absolutely stable. It can choose a large time step to improve the efficiency of calculation and satisfy the invariance of digital rotation.

The time parallel computing method is described below. P-M equation is time-dependent. Due to the irreversibility of time evolution, time-dependent equation can only be solved by strict time iteration algorithms. Until 2001, Lions and others proposed a real-time parallel algorithm based on time decomposition for solving time-depended problems (parareal algorithm) [13, 24, 25]. From this method was firstly proposed to today, it has been widely studied and applied in a few years. The algorithm decomposes the unsteady phase of Eulerian time differential system and solves the problem iteratively on the rough and fine grids of different scales, respectively.

Here is a brief introduction to the parareal algorithm [13], including some improvements to make it suitable for the problems we need to calculate. Suppose the following time-dependent equations are generally considered:

$$\begin{cases} \frac{\partial u}{\partial t} = Du, t \in [t_0, t_1], \\ u(t_0) = u(0). \end{cases} \quad (14)$$

For the above differential system,  $D$  is a differential operator independent of time differential, either linear or nonlinear. For example, for the two-dimensional heat conduction equation, the linear operator  $D = a(u_{xx} + u_{yy})$ , where  $a$  is a constant greater than 0.

For the above model Equation (14), the basic idea of time parareal is to divide the whole calculation time into several subintervals and then calculate in each subinterval at the same time. The basic algorithm is as follows:

- (i) Step 1: given precision  $\varepsilon > 0$ , with the idea of domain decomposition, the time interval  $[t_0, t_1]$  is divided into the following equidistant subintervals (rough time grids):

$$\Gamma = \{t_0 = T_0 < T_1 \cdots < T_n < T_{n+1} \cdots < T_{N_t-1} < T_{N_t} = t_1\}, \quad (15)$$

For convenience, let  $\Delta T = T_{n+1} - T_n$ . For the whole time interval  $[t_0, t_1]$ , the solution defined in the whole interval can be transformed into  $N$  subproblems defined in each subinterval  $[T_n, T_{n+1}]$ ,  $n = 1, 2, \dots, N_t$

- (ii) Step 2: based on the rough time grids  $\Gamma$ , take the explicit or implicit serial methods to calculate the sketchy approximate of the whole time points  $u(T_n)$ :  $u_1, u_2, \dots, u_n, \dots, u_{N_t-1}, u_{N_t}$ . For the convenience of the discussion, the calculation format on  $\Gamma$  can be written as follows:

$$u_{n+1} = C_{\Delta t}(u_n), \quad n = 1, 2, \dots, N_t, \quad (16)$$

where  $C_{\Delta t}$  is a numerical calculation format, for example, implicit or explicit.

- (iii) Step 3: subdivide each subinterval  $[T_n, T_{n+1}]$  into fine intercells:

$$T_n = T_n^0 < T_n^1 \cdots < T_n^i < T_n^{i+1} \cdots < T_n^s = T_{n+1}, \quad (17)$$

where  $s$  is number of the points in every subinterval  $[T_n, T_{n+1}]$ .

- (iv) Set the time interval length after subdivision is  $\delta t = T_n^{i+1} - T_n^i$ . Assign the value  $\{u_n\}$ ,  $n = 1, 2, \dots, N_t$  obtained in the second step to  $n$  subintervals as the initial condition of each interval. Solve parallelly Equation (14) in each intercells  $\{[T_{n-1}, T_n]\}$ ,  $n = 1, 2, \dots, N_t$ :

$$u_n^1, u_n^2, \dots, u_n^i, u_n^{i+1}, \dots, u_n^s, \quad n = 1, 2, \dots, N_t. \quad (18)$$

We can write the calculation format as follows:

$$u_n^{i+1} = F_{\delta t}(u_n^i), \quad i = 1, 2, \dots, s. \quad (19)$$

- (v) Step 4: correct the solution  $\{u_n^i\}$  on the rough grids by combining the solution obtained on the fine grids and then finish one-time iterative solution:



$$u_{n+1}^{(k+1)} = C_{\Delta t} \left( u_n^{(k+1)} \right) + F_{\delta t} \left( u_n^{(k)} \right) - C_{\Delta t} \left( u_n^{(k)} \right), \quad k = 0, 1, 2, \dots, M, \quad (20)$$

where the superscripts  $(k+1)$  and  $(k)$  are the number of iterative correction, and  $M$  is maxim correction steps.

- (vi) Step 5: if the error meets the accuracy,  $\|u^{(k+1)} - u^{(k)}\| < \varepsilon$ , the iteration will be stopped; otherwise, the iteration will go to the second step to continue the iteration

Note: in order to realize parallel computing quickly, rough and fine time grids are divided before each iteration. In each iteration, the results calculated in the last iteration need to be transmitted to the corresponding computing cells.

This paper combines AOS algorithm based on P-M model with time parareal method to denoise image. First of all, AOS is used to solve the approximate value on the rough time steps, since the fine intercell length is generally small after subinterval is subdivided, which can meet the requirements of the explicit method. Thus, explicit format can be used for parallel calculation on each subinterval, to increase the calculation accuracy and speed. The main discrete formats of our algorithm on denoising are listed below.

First, the implicit difference dispersion of Equation (3) is obtained

$$I_{i,j}^{n+1} = I_{i,j}^n + \tau \cdot \left( \frac{g_{i-1,j}^n + g_{i,j}^n}{2} \cdot (I_{i-1,j}^{n+1} - I_{i,j}^{n+1}) + \frac{g_{i+1,j}^n + g_{i,j}^n}{2} \cdot (I_{i+1,j}^{n+1} - I_{i,j}^{n+1}) + \frac{g_{i,j-1}^n + g_{i,j}^n}{2} \cdot (I_{i,j-1}^{n+1} - I_{i,j}^{n+1}) + \frac{g_{i,j+1}^n + g_{i,j}^n}{2} \cdot (I_{i,j+1}^{n+1} - I_{i,j}^{n+1}) \right). \quad (21)$$

The AOS methods (12) and (13) are used to solve the above discrete scheme (21). The initial guess values obtained on the rough time grids are set as the initial value of intercell parallel computing.

Next, on the fine grids, use the following explicit scheme for calculation:

$$I_{i,j}^{n+1} = I_{i,j}^n + \lambda \cdot (g_N^n \cdot \nabla_N^n I + g_S^n \cdot \nabla_S^n I + g_E^n \cdot \nabla_E^n I + g_W^n \cdot \nabla_W^n I). \quad (22)$$

Furthermore, use the formula to correct the previous two calculations:

$$u_{n+1}^{(k+1)} = C_{\Delta t} \left( u_n^{(k+1)} \right) + F_{\delta t} \left( u_n^{(k)} \right) - C_{\Delta t} \left( u_n^{(k)} \right). \quad (23)$$

Take the corrected results as the initial value of the next iteration. Computing in this way, we continue to iterate until reach the ideal degree of accuracy.

Note: parareal algorithm is very efficient for time-dependent equation(s). Compared with traditional domain parallel methods, the remarkable feature of this algorithm is its time parallel. Many test examples show that the algorithm has fast convergence, high efficiency, and easy to pro-

gram. It is also worth noting that if one can consider the time variable as a special spatial variable, he can use the domain decomposition algorithm based on spatial variables to calculate the change of time dimension. Therefore, in this sense, time parareal algorithm is also a domain decomposition algorithm, which is just a nonoverlapping domain decomposition algorithm.

### 3. Results and Discussion

In order to test the denoising effect of the above-mentioned parallel algorithm based on iteration, the time explicit method, the implicit method, and the time parareal computing technique of P-M equation are used to denoise the noisy image. In order to evaluate the denoising effect, we introduce the mean squared error (MSE) to assess the approximation degree to original clear image and choose termination index of the denoising. The mean square deviation measure of image is given by the following formula:

$$MSE = \sum_{i=1}^M \sum_{j=1}^N \left[ \frac{I(i,j) - \hat{I}(i,j)}{M_x \cdot N_y} \right]^2 \quad (24)$$

In the above formula (24),  $I(x,y)$  is the noisy image,  $\hat{I}(x,y)$  is the denoised image,  $M_x, N_y$  are the number of pixels in the  $x, y$  direction of the image. Based on the above definition, the smaller the mean square deviation measure is, the closer the denoised image is to the original image, the better of the denoising effect is.

For the sake of simplicity, firstly, we choose a gray image to illustrate the effect of denoising based on time parareal and P-M model. Since the noise of the actual image can be approximated by Gaussian noise, here, we add some Gaussian noise to the original image to get the noisy image. The first part (part a) in Figure 1 is the selected actual image of  $6000 \times 4000$  pixels. The b part is a noisy image-attached normal Gaussian noise with standard deviation of 15. Part c in Figure 1 is the result of denoising image based on the time parareal method and P-M model.

Figure 2 shows the different distribution in strong contrast and low contrast area of Gaussian noise image with variance of 15. Since the variance of noise is 15, it is relatively small compared with the maximum value 255 of gray image, so there is obvious noise pollution in the light part of image. In the light and dark part of the two small images on the right, the influence of noise can also be seen clearly. This kind of influence is similar to the image we take in the dark scene or high ISO light sensitive situation, so this kind of experiment has a good representativeness in real image capture and processing.

The following Figure 3 is the MSE change curve depended on time of the above-mentioned landscape denoising using the explicit, implicit, and time parareal algorithm. The abscissa is the time, and the ordinate represents the value of MSE. The top curve represented by a circle in the figure comes from the time parareal algorithm, the middle curve represented by a plus sign comes from the implicit serial algorithm, and the next curve with a small rectangle

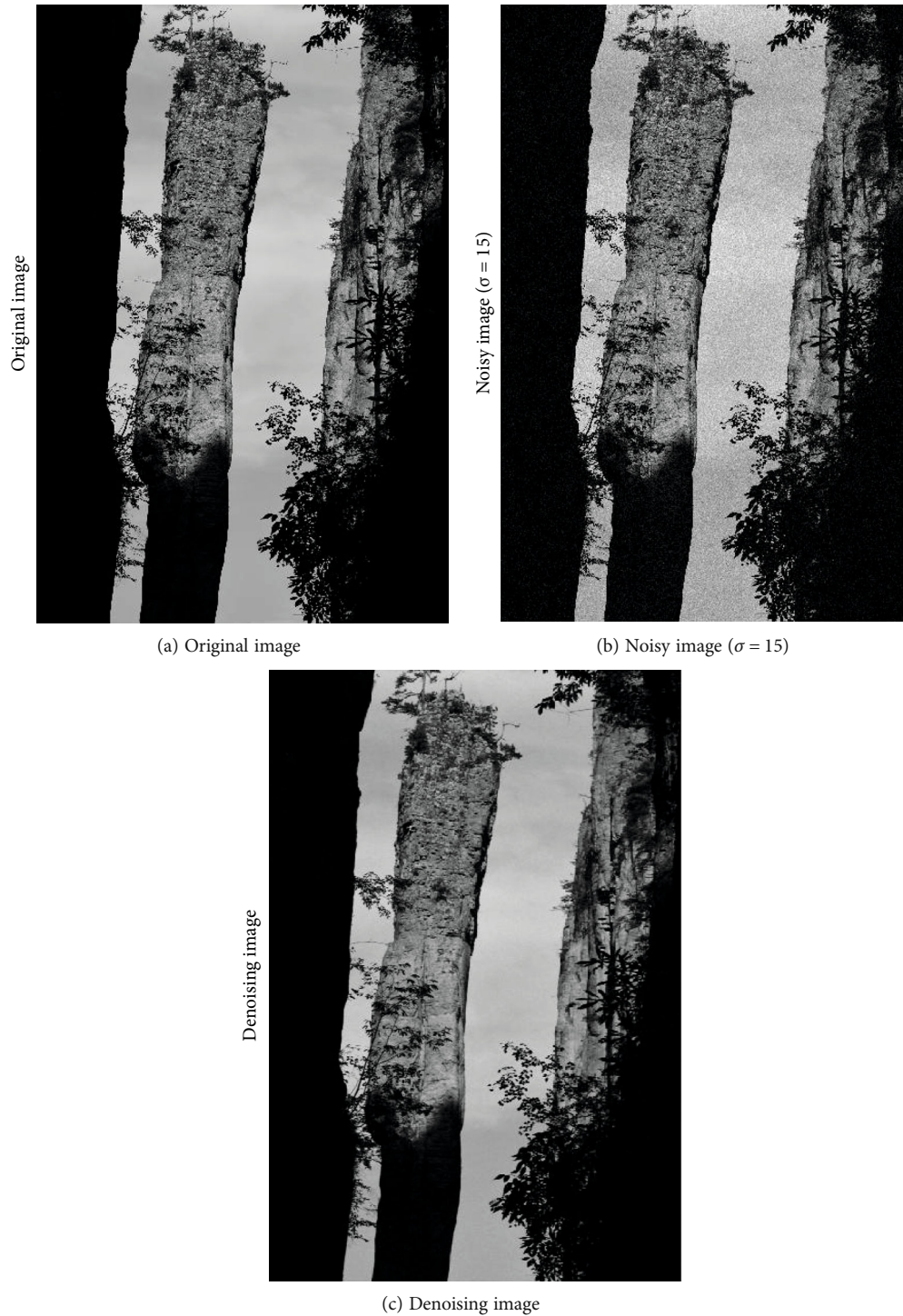


FIGURE 1: Original image, noisy image, and denoising image.

comes from the time serial algorithm. In order to ensure the accuracy and convergence of denoising, the time step of explicit scheme is 0.0005, and maxim iteration step is 4000; the time step of implicit scheme is 0.05 times, and the result of 400 iterations is adopted; the parareal calculation scheme is divided the interval  $[t_0, t_1]$  into 100 subintervals and adopts implicit scheme, each subinterval is divided into

120 small intercells, and 3 rounds of parallel iteration is finished.

Note: the denoising method based on PDE generally achieves a better denoising effect after appropriate number of time iteration, but the curve of time parareal algorithm is not a simple curve changing with time, because the time parareal algorithm is paralleled between regions and the

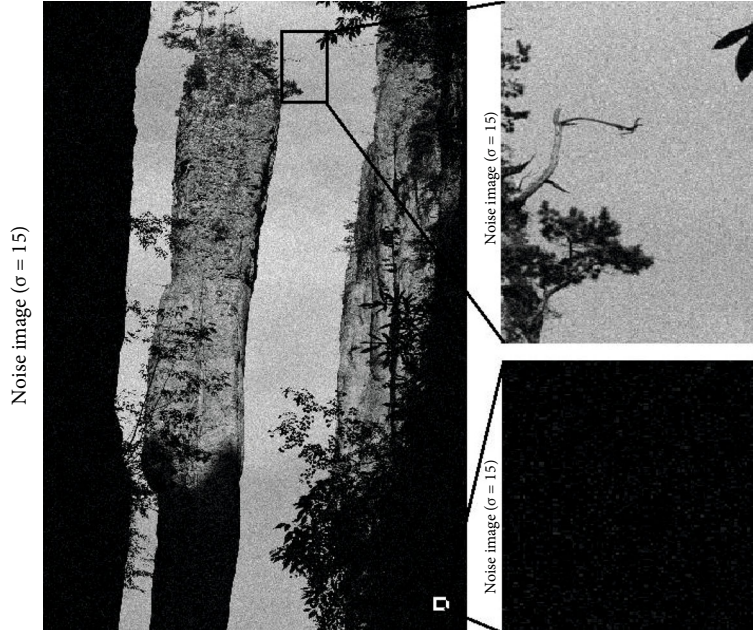


FIGURE 2: Different distribution of Gaussian noise image ( $\sigma = 15$ ).

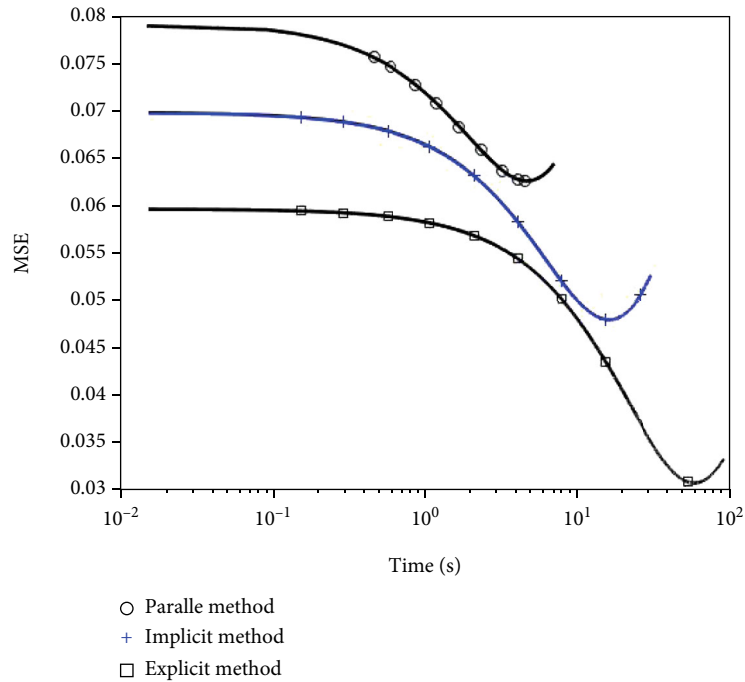


FIGURE 3: MSE comparison of time parareal method and explicit and implicit method.

TABLE 1: Efficiency comparison for different methods.

	Min (MSE)	Time (s)	Iterations
Explicit method	0.306647	57.44969	3794
Implicit method	0.478135	16.06848	359
Parareal method	0.624097	5.051357	—

converging to the original equation is a whole process, and does not simply come from the sample data after the parallel algorithm converges with time. So, the solution here is to select a parallel computing result after global convergence as the representative of the solution in different time steps and then make these computing results evenly distributed on the computing time axis. So, the time axis here is the last round of parallel iteration, that is, the change of MSE with time in the third round (see the circle-solid curve in the following Figure 2).



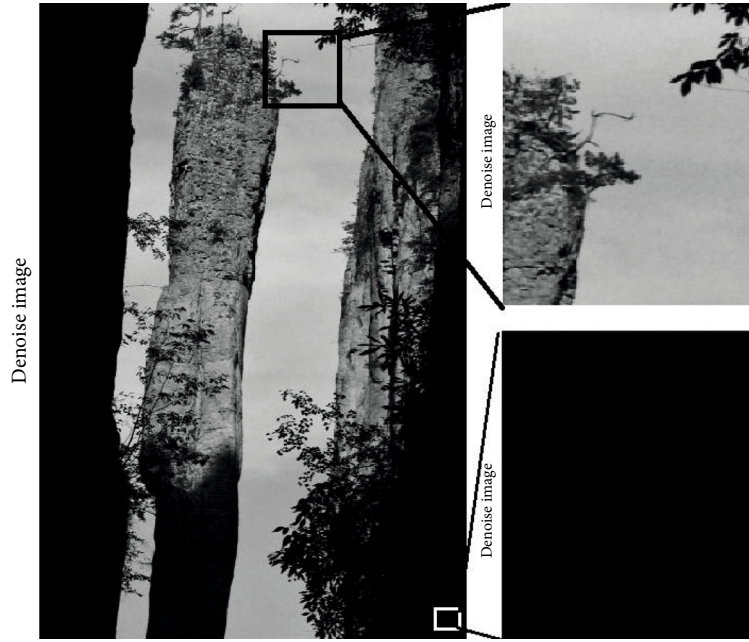


FIGURE 4: Denoising effect of Gaussian noise overlay ( $\sigma = 15$ ).

It can be seen from Figure 2 that, in general, the algorithm based on time parareal can save computing time effectively when the computing hardware resources (100 computing cells/processes) are ignored. The implicit format takes a relatively large time step. Although each iteration needs to solve two tridiagonal equations, the overall time spent is relatively small. However, the explicit serial computing is relatively simple; since a relatively small step is taken here, the time spent is actually the longest one and not dominant among the three methods. The following table shows the minimal MSE value and corresponding the cost time and iterations of explicit format, implicit format, and time parareal format.

From Table 1, we can get intuitively, the smaller the MSE, the closer the denoised image is to the original image, the better the denoising effect. Besides, because the explicit method requires a strict time step and there are many iterations needed, the iteration takes a lot of time. It takes about 1 minute to reach the minimum value, but the MSE values are generally smaller, and the MSE minimum computed is also the smallest one. Although the implicit method can take a larger step size, in order to ensure the effect of denoising, the step size should not be too large, which requires less time than the explicit method and is relatively stable. For the parareal algorithm, in theory, if the number of iterations is enough, the convergence of the calculation is determined by the step size of the fine grids. However, because the number of iterations is relatively small (only 3 turns), the MSE value is also affected by the step size of the rough grids. In this way, the MSE is the largest of the three, but the time is really saved.

Figure 4 is a denoised image based on the time parareal method. The effect is better in the distribution of strong contrast and low contrast than the local mean method mentioned in the introduction. Although the time parareal

MSE is larger than the explicit and implicit schemes, the  $L_2$  norm represents the macrodistribution level of noise, so there is little difference in the macroperformance of the image after denoising. In fact, for small-scale images, for the same scale of noise level, the explicit, implicit, and time parareal method cost the same time to denoise, they can also achieve the same noise level. Furthermore, if we choose different steps to calculate, the difference is not conspicuous.

In addition, the three discrete methods, in general, smoothen the image features in different degrees while smoothing the noise, but pay attention to the twigs on the rock, based on P-M method; the processing effect is relatively good at the noise level of variance 15.

Here are also three points worth noting: first, large-scale parallel computing devices do not seem to be often available, but the development of cloud computing and edge computing provides more possibilities for image parallel computing. Secondly, generally speaking, the convergence of time parareal algorithm is related to the governing equation. If the equation is highly nonlinear, the time step should be relatively small, or a more stable method should be used, such as implicit method or multistep method. Finally, because the calculation range of time is usually uncertain, we cannot deal with time parallelism completely according to the spatial parallel algorithm, which is also different from spatial parallelism.

#### 4. Conclusions

Based the above discussion of the classical nonlinear diffusion equation, P-M model, a fast denoising method is obtained by combining the unconditionally stable explicit and implicit schemes with the time parareal computing. Experimental results show that the parallel algorithm can effectively remove the noise while maintaining the image



edge information. Compared with the traditional explicit method, the effect of noise removal is not as good as that of implicit method, and the running time is far less than that of implicit method. It is about a few percent of the implicit scheme. Therefore, the image denoising algorithm based on time parareal and P-M model is an efficient and feasible numerical scheme, which can be applied to large-scale image denoising and has a good application prospect.

### Data Availability

No data were used to support this study.

### Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

### Acknowledgments

The author is very grateful for the inspiration of relevant literature and the valuable opinions put forward by the reviewers. The author thanks the research team of computing science and engineering for their enlightening suggestions in the process of study. This research was funded by the National Science Foundation of China, grant numbers: 10926133 and 60973015.

### References

- [1] A. Witkin, "Scale-space filtering," in *International Joint Conference on Artificial Intelligence*, pp. 1021–1091, William Kaufmann Publishers Inc, Karlsruhe, 1983.
- [2] C. Kalyoncu, C. Kalyoncu, and Ö. Toygar, "A weighted mean filter with spatial-bias elimination for impulse noise removal," *Digital Signal Processing*, vol. 46, pp. 164–174, 2015.
- [3] P. Hsieh, P. Shao, and S. Yang, "A regularization model with adaptive diffusivity for variational image denoising," *Signal Processing*, vol. 149, pp. 214–228, 2018.
- [4] W. Zhu and T. Chan, "Image denoising using mean curvature of image surface," *SIAM Journal on Imaging Sciences*, vol. 5, no. 1, pp. 1–32, 2012.
- [5] C. Sutour, C. A. Deledalle, and J. F. Aujol, "Estimation of the noise level function based on a nonparametric detection of homogeneous image regions," *SIAM Journal on Imaging Sciences*, vol. 8, no. 4, pp. 2622–2661, 2015.
- [6] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1-4, pp. 259–268, 1992.
- [7] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.
- [8] Y. Shen, Q. Liu, S. Lou, and Y. L. Hou, "Wavelet-based total variation and nonlocal similarity model for image denoising," *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 877–881, 2017.
- [9] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising prior driven deep neural network for image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 10, pp. 2305–2318, 2019.
- [10] J. Xu, L. Zhang, and D. Zhang, "External prior guided internal prior learning for real-world noisy image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2996–3010, 2018.
- [11] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, 2000.
- [12] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [13] A. K. Jain, "Partial differential equations and finite-difference methods in image processing, part 1: image representation," *Journal of Optimization Theory and Applications*, vol. 23, no. 1, pp. 65–91, 1977.
- [14] J. Weickert, "Applications of nonlinear diffusion in image processing and computer vision," *Acta Mathematica Universitatis Comenianae*, vol. 1, pp. 33–50, 2000.
- [15] Y. Y. Shih, J. C. Chen, and R. S. Liu, "Development of wavelet de-noising technique for PET images," *Computerized Medical Imaging and Graphics*, vol. 29, no. 4, pp. 297–304, 2005.
- [16] D. Gabor and J. Malik, "Information theory in electron microscopy," *Laboratory Investigation*, vol. 14, pp. 801–807, 1965.
- [17] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. on PAMI*, vol. 12, no. 7, pp. 629–639, 1990.
- [18] J. L. Lions, Y. Maday, and G. Turinici, "A "parareal" in time discretization of PDE's," *CR Acad. Sci. I – Math.*, vol. 332, no. 7, pp. 661–668, 2001.
- [19] J. Cortial and C. Farhat, "A time-parallel implicit method for accelerating the solution of non-linear structural dynamics problems," *International Journal for Numerical Methods in Engineering*, vol. 77, no. 4, pp. 451–470, 2009.
- [20] P. Amodio and L. Brugnano, "Parallel solution in time of ODEs: some achievements and perspectives," *Applied Numerical Mathematics*, vol. 59, no. 3-4, pp. 424–435, 2009.
- [21] F. Catté, P. L. Lions, J. M. Morel, and T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM Journal on Numerical Analysis*, vol. 29, no. 1, pp. 182–193, 1992.
- [22] J. Weickert and H. Scharr, "A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance," *Journal of Visual Communication and Image Representation*, vol. 13, no. 1-2, pp. 103–118, 2002.
- [23] J. Xu, "A novel two-grid method for semilinear elliptic equations," *SIAM Journal on Scientific Computing*, vol. 15, no. 1, pp. 231–237, 1994.
- [24] C. Farhat, J. Cortial, C. Dastillung, and H. Bavestrello, "Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses," *International Journal for Numerical Methods in Engineering*, vol. 67, no. 5, pp. 697–724, 2006.
- [25] G. Frantziskonis, K. Muralidharan, P. Deymier, S. Simunovic, P. Nukala, and S. Pannala, "Time-parallel multiscale/multi-physics framework," *Journal of Computational Physics*, vol. 228, no. 21, pp. 8085–8092, 2009.