mathematics

Article

Overlapping Domain Decomposition Method with Cascadic Multigrid for Image Restoration

Zhaoteng Chu and Chenliang Li *

School of Mathematics and Computing Science, Center for Applied Mathematics of Guangxi (GUET), Guangxi University Key Laboratory of Data Analysis and Computation, Guilin University of Electronics Technology, Guilin 541004, China

* Correspondence: chenli@guet.edu.cn

Abstract: In the process of image restoration, it is usually necessary to solve large-scale inverse problems, where the computational cost is very high for large or high-resolution images. The domain decomposition method is one of the most effective algorithms to solve large-scale problems, which can effectively decrease the computational cost. The cascadic multigrid method has a good effect on the linear model of image restoration and can obtain high quality restored images. In this paper, the overlapping domain decomposition method (DDM) with the cascadic multigrid method (CMG) and the DDM with new extrapolation cascadic multigrid method (NECMG) are presented to solve the image restoration problems of denoising and deblurring. We first divide the image problem into some overlapping and independent subproblems. Then, each subproblem is solved independently by CMG or NECMG with the edge-preserving operator. Numerical experiments show that the new method is effective.

Keywords: overlapping domain decomposition method; cascadic multigrid method; new extrapolation cascadic multigrid method; edge preserving; denoising and deblurring; reflexive

MSC: 65N20; 65N55; 65Y05; 68U10

Citation: Chu, Z.; Li, C. Overlapping Domain Decomposition Method with Cascadic Multigrid for Image Restoration. *Mathematics* **2023**, *11*, 2389. https://doi.org/10.3390/math11102389

Academic Editor: JAVIER MARTÍNEZ TORRES

Received: 5 April 2023
Revised: 17 May 2023
Accepted: 17 May 2023
Published: 22 May 2023

1. Introduction

In many applications of image processing, the solution of the large-scale linear discrete ill-posed equation or the least squares solution are required. Denoising and deblurring are some of the most important and challenging tasks in image restoration. For recent works on image denoising and deblurring for large scale linear discrete ill-condition equations and least squares, see [1–3].

Consider the restoration of two-dimensional gray-scale images [4],

$$\hat{f}(x) = \int_{\Omega} h(x,y)\hat{u}(y)dy + \eta(x), \tag{1}$$

where $\hat{f}(x)$ is a blurry and noisy image, $h(x,y)$ is the point spread function (PSF), $\eta(x)$ represents additive noise in the available data $f(x)$, and $\hat{u}(y)$ is the clean image.

The discretized form of Equation (1) can be expressed as

$$f = H\hat{u} + e, \tag{2}$$

where the noise is obtained by adding random a normally distributed vector e with mean 0 and standard deviation 1, and $\delta = \|e\|$. H represents the fuzzy operator, which is usually a real symmetric block Teoplitiz matrix, $\hat{u}$ is the vector generated by the original image, and f represents the blur- and noise-contaminated image vector.

We would like to determine an approximate value of $\hat{u}$ when the blurry and noisy image $\hat{f}$ and the fuzzy operator H are known but the noise vector e is unknown. Ignoring the noise vector e, Equation (2) can be expressed as,

$$Hu = f, \tag{3}$$

where u is the solution we want to find.

Since the matrix H is a large-scale singular matrix, the linear discretization Equation (3) is usually an ill-posed problem. Although we can solve Equation (3) directly by some iterative methods, due to the loss of high-frequency information, these iterative methods cannot accurately recover edges, which seriously affects the quality of image restoration.

In order to determine meaningful values of $u(y)$, one of the classical methods of replacing Equation (2) is the Tikhonov regularization model,

$$\min_u \left\{ \int_\Omega \left(\frac{1}{2} \left(\int_\Omega h(x,y)u(y)dy - \hat{f}(x) \right)^2 + \alpha\psi(u(x)) \right) dx \right\}, \tag{4}$$

where $\alpha > 0$ is a regularization parameter and $\psi(u(x))$ is a regularization operator. The choice of different regularization operators creates different image restoration models.

For the nonlinear Equation (4), refs. [5,6] propose the explicit and semi-implicit discretization schemes, which have a good preserving effect on the edges of images and can provide high quality restoration of images; however, the amount of computation is extremely high. In addition, the selection of regularization parameters and time intervals is not easy.

In [4,7], the restoration Equation (3) is discussed in detail by using the cascadic multigrid method, which is effectively applied to the restoration of small- and medium-sized images. For large-scale images, linear and nonlinear often require a lot of storage space and computation time, which make the recovery process very difficult. In recent years, in order to solve these problems, many scholars have extended domain decomposition methods [8] to image restoration, achieving remarkable results for many kinds of restoration models.

Ref. [9] proposes a simple domain decomposition algorithm based on the waveform relaxation of nonlinear diffusion equations for denoising images. It shows the feasibility of the domain decomposition algorithm in image restoration. Ref. [10] proposes parallel overlapping domain decomposition methods and shows the successful application of the algorithm for the restoration of 1D signals and 2D images in interpolation/inpainting problems, but the CPU time of this method is very long. Refs. [11,12] combine the domain decomposition method with the graph cuts algorithm for total variation minimization. This method improves the computation efficiency and greatly reduces the memory cost and enables us to solve very large-scale image problems effectively. Ref. [13] combines the domain decomposition method and the Bregman algorithm for image inpainting and image deblurring. It accelerates the computation of subproblems by the nested Bregman iteration. It turned out that the proposed new solution is up to three times faster than the iterative algorithm currently used in domain decomposition methods. In [14], the meshfree finite point method with parallel domain decomposition is investigated for image denoising. The domain decomposition technique, which can be implemented in parallel, decreases the size of the system of equations and significantly reduces the computational cost. In [15,16], a fast algorithm based on overlapping domain decomposition technology is proposed to solve larger image problems and perform more efficiently than traditional method in terms of the restored image quality and CPU time. In [17], the image is denoised by the overlapping domain decomposition method based on the successive subspace correction method and parallel subspace correction method. This method shows the ability of processing large-scale images. Ref. [18] presents a fast domain decomposition method for global image smoothing. This method converges quickly after a few iterations and the runtime is much shorter than conventional methods. Ref. [19] presents a parallel domain decomposition algorithm for large scale image denoising, which, compared with [10], greatly reduces the calculation time.

The above methods have successfully solved the problem of large-scale image processing, but most of them only denoise or deblur the input image. For large-scale images that contain both noise and blur, we research a kind of quick and effective method.

The cascadic multigrid method is a unidirectional computation method from the coarse grid level to the fine grid level. The solution to the coarse grid provides a better initial value for the fine grid by interpolation methods, during which coarse grid correction is not required. The solution of the fine grid is obtained only after interpolation and iteration operations are used. The overlapping domain decomposition method has a strong performance in dealing with large-scale problems. Ref. [20] introduces a new parallel cascadic multigrid method. Ref. [21] introduces the multigrid domain decomposition method. Inspired by their methods, combining the overlapping domain decomposition method with the cascadic multigrid method, we propose a new algorithm for image restoration. Our algorithm has the following advantages:

(1) The algorithm can not only improve the computational efficiency and achieve good denoising and deblurring effects but also protect the image edge.

(2) The algorithm has a significantly faster convergence compared to when solving the problem directly at the finest level. The image is subdivided into multiple overlapping subimages, and each of the subimages is solved by the cascadic multigrid method or the new extrapolation cascadic multigrid method.

(3) The algorithm is parallel, which greatly saves storage cost and improves the computing efficiency.

The structure of this paper is as follows. In Section 2, we introduce the domain decomposition method with cascadic multigrids and the domain decomposition method with new extrapolated cascadic multigrids. Section 3 introduces the form and application of the edge-preserving denoising operator. Section 4 introduces how to locally weight the overlapped part of the subdomain. In Section 5, some numerical results are given. In Section 6, a summary is presented.

2. Overlapping Domain Decomposition Method with Cascadic Multigrid

In this section, we propose an overlapping domain decomposition method with the cascadic multigrid algorithm to solve the problem of image deblurring and denoising. The advantages of this method are that it has a lower cost and can be extended on parallel computers.

First, we divide the image domain into N non-overlapping subdomains denoted as $\Omega = \cup_{i\in\mathcal{I}}\Omega_i$ with $\mathcal{I} := \{1, 2, \ldots, N\}$ and $\Omega_i \cap \Omega_j = \emptyset, i \neq j$. Then, we extend each subdomain into a larger subdomain Ω_i^ρ, including ρ columns or ρ rows pixels from its neighboring subdomains. We can denote the overlapping size ρ as the distance between the boundaries of Ω_i and Ω_i^ρ. For any $i \in \mathcal{I}$, we define the set of neighboring subdomains as $\mathcal{K}_i := \left\{k \in \mathcal{I} : \Omega_i^\rho \cap \Omega_k^\rho \neq \emptyset, k \neq i\right\}$. For any $i \in \mathcal{I}$, $Q_i^{\mathcal{K}_i} := \cup_{k\in\mathcal{K}_i}\left(\Omega_i^\rho \cap \Omega_k^\rho\right)$ denotes the overlapping domain. $\Omega_i^\rho \setminus Q_i^{\mathcal{K}_i}$ denotes the non-overlapping domain. See Figure 1 for an example.

Then, we apply the domain decomposition method to Equation (3). We can parallelly solve the following subproblems,

$$\begin{cases} H_i u_i^n = f_i, & \text{in } \Omega_i^\rho, \\ u_i^n = u^{n-1}, & \text{on } \partial\Omega_i^\rho. \end{cases} \tag{5}$$

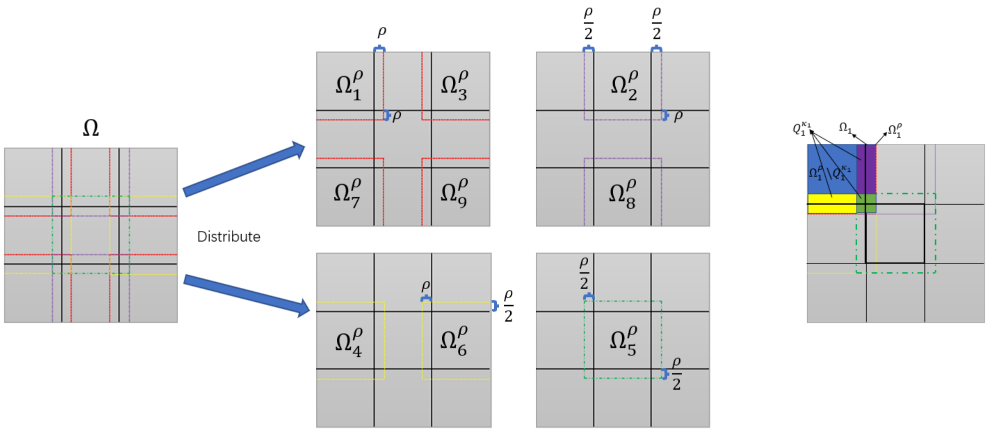


Figure 1. Overlapping domain decomposition.

Let us denote the solution of Equation (5) as $\tilde{u}_i^n$. After solving the problem on each subdomain, we will carry out the weighted processing (Equation (6)) for overlapped parts. More details will be given in Section 4.

$$u_i^n = \begin{cases} \tilde{u}_i^n, & \Omega_i^\rho \backslash Q_i^{\mathcal{K}_i}, \\ \theta\tilde{u}_i^n + \frac{(1-\theta)}{C_i^{\mathcal{K}_i}-1} \sum\limits_{\substack{k\in\mathcal{K}_i \\ \tilde{u}_k^n\in\Omega_k^\rho}} \tilde{u}_k^n, & Q_i^{\mathcal{K}_i}. \end{cases} \tag{6}$$

Next, we obtain the iteration solution u^n in the following way,

$$u^n = \sum_{i=1}^{N} E_i R_i u_i^n, \tag{7}$$

where R_i is a restriction operator from Ω_i^ρ to Ω_i and E_i is an extension operator from Ω_i to Ω. It can extend the subdomain to the entire domain. Either R_i or E_i is a matrix with only elements 0 and 1.

Then, the above process is repeated until the following stopping rules were satisfied,

$$\frac{\|u^n - u^{n-1}\|_2}{\|u^n\|_2} < \epsilon, \tag{8}$$

where $\epsilon > 0$ and is a given constant.

The framework of the domain decomposition method is given in Algorithm 1.

In Step (1), we can also solve the subproblem $H_i u_i = f_i$ by the cascadic multigrid method.

For the sake of discussion, we let every u_i be an $m^2 \times 1$ column vector, let m be an integer, and set $M = m^2$. Let $W_1 \subset \cdots \subset W_l \subset \cdots \subset W_L$ be a nested subspace. The subscript l indicates the number of the grid level, W_1 denotes the coarsest grid level, and W_L indicates the finest grid level. Let $u_i^{(l)}$ be the representation of u_i in W_l $(l = 1, \ldots, L)$. Accordingly, we have $M_1 < \cdots < M_l < \cdots < M_L = M$. When M is odd, $M_{l-1} = \frac{(\sqrt{M_l}+1)^2}{4}$; when M is even, $M_{l-1} = \frac{1}{4}M_l$.

Algorithm 1: Overlapping domain decompsition method (DDM) for image restoration

Input: u^0 , N , H, f.
Output: $u = u^n$.
for n $= 1, \ldots,$ *until convergence* **do**
  (1) Solve the system $H_i u_i^n = f_i$ with mimimum residual method:
    $\tilde{u}_i^n := MR(H_i, f_i), i = 1, \ldots, N$.
  (2) Get u_i^n by Equation (6).
  (3) $u^n = \sum_{i=1}^N E_i R_i u_i^n$.
  (4) **if** $\frac{\|u^n - u^{n-1}\|_2}{\|u^n\|_2} < \epsilon$ **then**
    $u = u^n$;
  else
    $n = n + 1$.
  end
end

In the cascadic multigrid algorithm, the following linear system needs to be solved iteratively at each W_l,

$$H_i^{(l)} u_i^{(l)} = f_i^{(l)}, \tag{9}$$

where $H_i^{(l)}$ is the representation of the fuzzy operator H_i in W_l.

Set $(f_i^{(l)})^{(k)}$ as the kth element of $f_i^{(l)}$, and we let

$$(f_i^{(l)})^{(k)} = (f_i^{(l+1)})^{(2k-1)}, k = 1, \ldots, M_l. \tag{10}$$

We define the restriction operator $R_i^{(l)}$ which satisfies

$$f_i^{(l)} = R_i^{(l)} f_i^{(l+1)} \tag{11}$$

when $l = L$, $f_i^{(L)} = f_i$.

$H_i^{(l)}$ is generated in the following way,

$$H_i^{(l)} = R_i^{(l)} H_i^{(l+1)} R_i^{*(l)}, \tag{12}$$

when $l = L$, $H_i^{(L)} = H_i$.

The prolongation operator $P_i^{(l)}$ is from level W_l to level W_{l+1} for each l. Generally, linear interpolation or quadratic interpolation is used as the prolongation operator.

We use $S_i^{(l)}$ to represent the smoother MR on level l in the i^{th} subdomain. The smoothing is terminated when the following stopping rule is satisfied.

Stopping rule: For a given $\delta \geq 0$, $c > 1$ is a constant and independent of δ. When

$$\|f_i^{(l)} - H_i^{(l)} u_i^{(l)}\| \leq c\delta, \tag{13}$$

the iteration is stopped. The number of iterations is denoted as $m_i^{(l)}$.

Thus, the cascadic multigrid method (CMG) in this paper first determines an approximate solution of $H_i^{(1)} u_i^{(1)} = f_i^{(1)}$ in W_1 using MR. Then, this iterative value is mapped from W_1 to W_2 by the prolongation operators $P_i^{(2)}$. A correction of this mapping in W_2 is iterated by MR and terminated with the stopping rule. The approximate solution $u_i^{(1)}$ in W_2 is mapped into W_3 by the prolongation operators $P_i^{(3)}$. The computations are continued in this manner until an approximate solution has been determined in the finest grid level W_L.

In our algorithm, quadratic interpolation is used as the prolongation operator $P_i^{(l)}$. The edge-preserving denoising operator is denoted as $D_i^{(l)}$. The framework of the domain decomposition method with cascadic multigrids is given in Algorithm 2.

Algorithm 2: Overlapping domain decomposition method with cascadic multigrid (DDM-CMG) for image restoration

Input: u^0 , N , L, H, f.
Output: $u = u^n$.
for n $= 1, \ldots,$ *until convergence* **do**
  (1) Solve the system $H_i u_i^n = f_i$ $(i = 1, \ldots, N)$ with CMG.
    Give the initial iteration value $u_i^{(0),n}$, we can get $u_i^{(1),n} := S_i^{(l),n}\ u_i^{(0),n}$.
  for $l = 2, \ldots, L$ **do**
    (a) Prolongation: $\bar{u}_i^{(l),n} := P_i^{(l),n}\ u_i^{(l-1),n}$;
    (b) Smoothing until Equation (13) holds: $\hat{u}_i^{(l),n} := S_i^{(l),n}\ \bar{u}_i^{(l),n}$;
    (c) Edge preserving denoising: $\tilde{u}_i^{(l),n} := D_i^{(l),n}\ \hat{u}_i^{(l),n}$;
  end
  (2) $\tilde{u}_i^n := \hat{u}_i^{(L),n}$.
  (3) Get u_i^n by Equation (6).
  (4) $u^n = \sum_{i=1}^N E_i R_i u_i^n$.
  (5) **if** $\frac{\|u^n - u^{n-1}\|_2}{\|u^n\|_2} < \epsilon$ **then**
    $u = u^n$;
  else
    $n = n + 1$.
  end
end

In [22], the new extrapolation cascadic multigrid method (NECMG) is proposed, which can provide better initial values for the next level and has better convergence by using new extrapolation formulas and quadratic interpolation.

Let us take the one-dimensional triplet grid $G_l (l = 1, 2, 3)$ as an example to understand the new extrapolation formulas and quadratic interpolation. The pixel at the i^{th} node on G_l is denoted as x_i^l, and the corresponding pixel value is denoted by u_i^l . Let I_l be the node cell on the grid level G_l and they are represented as follows,

$$I_1 = (x_i^1, x_{i+1}^1)\ ,\ I_2 = (x_i^2, x_{(i+1)/2}^2, x_{i+1}^2)\ ,\ I_3 = (x_i^3, x_{(i+1)/4}^3, x_{(i+1)/2}^3, x_{(i+3)/4}^3, x_{i+1}^3).$$

The value at the corresponding node can be expressed as,

$$G_1 = (u_i^1, u_{i+1}^1)\ ,\ G_2 = (u_i^2, u_{(i+1)/2}^2, u_{i+1}^2)\ ,\ G_3 = (u_i^3, u_{(i+1)/4}^3, u_{(i+1)/2}^3, u_{(i+3)/4}^3, u_{i+1}^3).$$

Our aim is to try to use the values at five nodes on G_1 and G_2 to provide a good initial value on the finest grid G_3.

(1) New extrapolation, $u_3 = \Pi_2(u_1, u_2)$.

$$\begin{cases} u_i^3 = (5u_i^2 - u_i^1)/4\,, \\ u_{i+1}^3 = (5u_{i+1}^2 - u_{i+1}^1)/4\,, \\ u_{i+1/2}^3 = u_{i+1/2}^2 + [(u_i^2 - u_i^1) + (u_{i+1}^2 - u_{i+1}^1)]/8\,. \end{cases} \tag{14}$$

(2) Quadratic interpolation, $u_3 = P_3(u_1, u_2, u_3)$.

$$\begin{cases} u^3_{i+1/4} = [(9u^2_i + 12u^2_{i+1/2} - u^2_{i+1}) - (3u^1_i + u^1_{i+1})]/16, \\ u^3_{i+3/4} = [(9u^2_{i+1} + 12u^2_{i+1/2} - u^1_i) - (3u^1_{i+1} + u^1_i)]/16. \end{cases} \tag{15}$$

Based on Algorithm 2 with new extrapolation formulas and quadratic interpolation, we present Algorithm 3.

In Algorithms 2 and 3, the smoothing is terminated also according to a stopping rule, see [7,23] for more information.

Algorithm 3: Overlapping domain decomposition method with new extrapolation cascadic multigrid (DDM-NECMG) for image restoration

Input: u^0 , N , L, H, f.
Output: $u = u^n$.
for $n = 1, \ldots,$ *until convergence* **do**
    (1) Solve $H_i u^n_i = f_i$ $(i = 1, \ldots, N)$ with NECMG.
    (i) Given the initial value $u^{(01),n}_i$ on $l = 1$ and $u^{(02),n}_i$ on $l = 2$, we get $u^{(1),n}_i := S^{(1),n}_i u^{(01),n}_i$, $u^{(2),n}_i := S^{(2),n}_i u^{(02),n}_i$.
    (ii) Use the new extrapolation operator we update $u^{(2),n}_i := \Pi^{(2),n}_i(u^{(1),n}_i, u^{(2),n}_i)$
    (iii) **for** $l = 3, \ldots, L$ **do**
        (a) prolongation: $\bar{u}^{(l),n}_i := P^{(l),n}_i \; u^{(l-1),n}_i$;
        (b) smoothing until Equation (13) holds: $\hat{u}^{(l),n}_i := S^{(l),n}_i \; \bar{u}^{(l),n}_i$;
        (c) edge preserving denoising: $\tilde{u}^{(l),n}_i := D^{(l),n}_i \; \hat{u}^{(l),n}_i$;
    end 2
    $\tilde{u}^n_i := \hat{u}^{(L),n}_i$.
    (3) Get u^n_i by Equation (6).
    (4) $u^n = \sum_{i=1}^{N} E_i R_i u^n_i$.
    (5) **if** $\frac{\|u^n - u^{n-1}\|_2}{\|u^n\|_2} < \epsilon$ **then**
        $u = u^n$;
    else
        $n = n + 1$.
    end
end

3. Edge-Preserving Denoising Operator

In this section, we introduce the edge-preserving denoising operator used in Algorithms 2 and 3.

In nonlinear model 4, the discrete regularization term $\psi(u(x))$ is denoted by $\Psi(u)$, where the appropriate selection of the regularization operator $\Psi(u)$ is crucial for the quality of image restoration. The most classical model is the following nonlinear anisotropic diffusion Equation (16),

$$\begin{cases} \Psi(u) = \mathrm{div}[z(|\nabla u|)\nabla u], \\ u(x, y, t)|_{t=0} = u_0(x, y), \end{cases} \tag{16}$$

where $u(x, y, t)$ is the image at moment t, $z(|\nabla u|) \in [0, 1]$ is the diffusion coefficient, and the diffusion coefficients are chosen as follows,

$$z(|\nabla u|) = \frac{1}{1 + \left(\frac{|\nabla u|}{k}\right)^2}, \quad k \text{ for the threshold.} \tag{17}$$

Set $c(x,y) = z(|\nabla u|)$, then the finite difference discrete scheme of Equation (16) can be expressed as:

$$\begin{aligned}\frac{u^{m+1}_{(s,j)} - u^{m}_{(s,j)}}{\tau} = & \frac{c^{m}_{(s+1,j)} + c^{m}_{(s,j)}}{2}\left(u^{m}_{(s+1,j)} - u^{m}_{(s,j)}\right) + \frac{c^{m}_{(s-1,j)} + c^{m}_{(s,j)}}{2}\left(u_{(s-1,j)} - u^{m}_{(s,j)}\right) \\ & + \frac{c^{m}_{(s,j+1)} + c^{m}_{(s,j)}}{2}\left(u^{m}_{(s,j+1)} - u^{m}_{(s,j)}\right) + \frac{c^{m}_{(s,j-1)} + c^{m}_{(s,j)}}{2}\left(u^{m}_{(s,j-1)} - u^{m}_{(s,j)}\right),\end{aligned} \tag{18}$$

where $u^{m}_{(s,j)}$ represents the pixel value at (s,j) in the finite difference scheme.

Equation (18) can be expressed in matrix form as ,

$$u^{m+1} = u^{m} + \tau A(u^{m})u^{m}, \tag{19}$$

where u^{m} and u^{m+1} respectively denote the m and $m+1$ time moment images.

Since the nonlinear Equation (19) has a good preserving effect on the edge of the image, we apply it to the restoration of linear Equation (3) as an edge-preserving denoising operator $(D_i^{(l)})$. Specifically, let us take $\tau = 0.1$ and implement Equation (19) for ten iterations.

4. Weight in the Overlapping Domains

In this section, we discuss the weight in the overlapping domains. In Section 2, we introduce Equation (6), as follows,

$$u_i^n = \begin{cases} \tilde{u}_i^n, & \Omega_i^{\rho} \backslash Q_i^{\mathcal{K}_i} \\ \theta \tilde{u}_i^n + \frac{(1-\theta)}{C_i^{\mathcal{K}_i}-1} \sum\limits_{\substack{k \in \mathcal{K}_i \\ \tilde{u}_k^n \in \Omega_k^{\rho}}} \tilde{u}_k^n, & Q_i^{\mathcal{K}_i} \end{cases},$$

where $Q_i^{\mathcal{K}_i}$ is defined as each distinct overlapping subdomain generated by $\Omega_i \cap \Omega_k^{\rho}$ for any $i \in \mathcal{I}$ and $k \in \mathcal{K}_i$. $C_i^{\mathcal{K}_i}$ is a constant which is defined as the number of different overlapping subdomains in $Q_i^{\mathcal{K}_i}$. For example, let us take $i = 1$, see Figure 2.

$$Q_1^{\mathcal{K}_1} \begin{cases} Q_1^{1,2}, & \Omega_1^{\rho} \cap \Omega_2^{\rho} \\ Q_1^{1,4}, & \Omega_1^{\rho} \cap \Omega_4^{\rho} \\ Q_1^{1,2,4,5}, & \Omega_1^{\rho} \cap \Omega_2^{\rho} \cap \Omega_4^{\rho} \cap \Omega_5^{\rho} \end{cases}, \quad C_1^{\mathcal{K}_1} \begin{cases} C_1^{1,2} = 2, & \Omega_1^{\rho} \cap \Omega_2^{\rho} \\ C_1^{1,4} = 2, & \Omega_1^{\rho} \cap \Omega_4^{\rho} \\ C_1^{1,2,4,5} = 4, & \Omega_1^{\rho} \cap \Omega_2^{\rho} \cap \Omega_4^{\rho} \cap \Omega_5^{\rho} \end{cases}.$$

Thus, we can obtain u_1^n in the subdomain Ω_1^{ρ},

$$u_1^n = \begin{cases} \tilde{u}_1^n, & \Omega_1^{\rho} \backslash Q_1^{\mathcal{K}_1} \\ \theta \tilde{u}_1^n + (1-\theta)\tilde{u}_2^n, & Q_1^{1,2} \\ \theta \tilde{u}_1^n + (1-\theta)\tilde{u}_4^n, & Q_1^{1,4} \\ \theta \tilde{u}_1^n + \frac{(1-\theta)}{3}(\tilde{u}_2^n + \tilde{u}_4^n + \tilde{u}_5^n), & Q_1^{1,2,4,5} \end{cases} \tag{20}$$

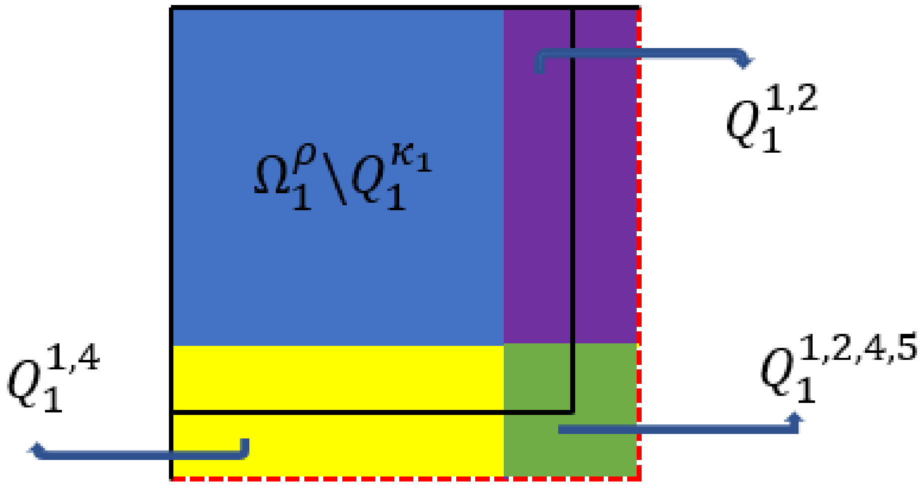


Figure 2. The overlapping domains $Q_1^{1,2}$, $Q_1^{1,4}$, and $Q_1^{1,2,4,5}$ and nonoverlapping domain $\Omega_1^{\rho}\backslash Q_1^{\mathcal{K}_1}$.

5. Numerical Examples

In this section, we give several numerical examples to show the effectiveness of our algorithms.

The elements in the fuzzy operator H_i are defined by the following function,

$$h_{(s,j)} = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(s-j)^2}{2\sigma^2}\right), & \text{if } |s-j| \leqslant \text{ band}, \\ 0, & \text{other}, \end{cases}$$

$$h_i = \left[h_{(s,j)}\right]_{s,j=1,\dots,M},$$

$$H_i = h_i \otimes h_i,$$

where $\otimes$ denotes the Kronecker product, M represents the dimensions of the subimage matrix, σ represents the variance of the Gaussian point diffusion function, and the value of band represents the half-bandwidth of the Toplitz matrix h_i. The larger the values of σ and band, the fuzzier the image generated.

The image restoration effect is measured by the following peak signal ratio,

$$\text{PSNR}(u^n, u) = 20 \log_{10} \frac{255}{\|u^n - u\|} \text{dB}.$$

The three original images are shown in Figure 3. Let $\sigma = 2$ and band=11; then, we obtain noise and blur images as shown in Figure 4. In our experiments, let $\epsilon = 10^{-4}$. We check the effectiveness of Algorithms 1–3.

(**a**) Pirate 1083×1083 (**b**) Fairy 783×783 (**c**) Lena 543×543

Figure 3. Original images.

(**a**) PSNR = 7.638 (**b**) PSNR = 12.2887 (**c**) PSNR = 9.459

Figure 4. Blurred and noisy images.

Tables 1–3 and Figures 5–7 show the numerical results of DDM, DDM-CMG, and DDM-NECMG. From Figure 8, compared with PSNR and CPU time, DDM-CMG and DDM-NECMG are superior to DDM, where DDM-NECMG is the best.

This shows that Algorithm 3 not only can achieve a good recovery effect in a very short time, but also solves large resolution images effectively.

Table 1. Numerical results of Pirate 1083 × 1083.

Pirate	DDM		DDM-CMG		DDM-NECMG	
	PSNR	**Time (s)**	**PSNR**	**Time (s)**	**PSNR**	**Time (s)**
Ω	14.111	8.776	16.401	7.400	16.504	4.325
Ω_1^ρ	23.167	0.781	25.198	0.796	24.851	0.438
Ω_2^ρ	22.999	0.781	24.948	0.656	25.208	0.312
Ω_3^ρ	21.769	0.937	23.745	0.562	23.995	0.374
Ω_4^ρ	23.106	0.781	25.062	0.796	24.637	0.374
Ω_5^ρ	22.113	0.783	24.026	0.656	23.862	0.374
Ω_6^ρ	22.578	1.093	24.742	0.609	24.821	0.312
Ω_7^ρ	22.927	0.937	24.795	0.750	24.949	0.312
Ω_8^ρ	22.212	0.468	24.324	0.656	24.662	0.374
Ω_9^ρ	21,966	1.250	23.857	0.515	24.119	0.312

Table 2. Numerical results of Fairy 783 × 783.

Fairy	DDM		DDM-CMG		DDM-NECMG	
	PSNR	**Time (s)**	**PSNR**	**Time (s)**	**PSNR**	**Time (s)**
Ω	18.302	8.275	19.808	5.937	20.147	3.225
Ω_1^ρ	29.451	0.625	31.004	0.437	31.177	0.218
Ω_2^ρ	25.356	0.312	26.858	0.562	26.136	0.218
Ω_3^ρ	29.280	1.562	30.770	0.312	29.992	0.218
Ω_4^ρ	23.763	0.937	25.245	0.375	25.704	0.250
Ω_5^ρ	25.313	0.312	27.306	0.625	27.329	0.187
Ω_6^ρ	26.720	0.937	28.573	0.500	28.649	0.218
Ω_7^ρ	26.191	0.937	27.621	0.562	27.818	0.187
Ω_8^ρ	26.022	0.625	27.742	0.062	27.849	0.218
Ω_9^ρ	27.985	0.937	29.687	0.500	29.858	0.218

Table 3. Numerical results of Lena 543 × 543.

Lena	DDM		DDM-CMG		DDM-NECMG	
	PSNR	**Time (s)**	**PSNR**	**Time (s)**	**PSNR**	**Time (s)**
Ω	20.749	3.433	23.752	2.656	23.806	2.329
Ω_1^ρ	28.740	0.156	32.038	0.234	30.400	0.187
Ω_2^ρ	27.682	0.312	30.767	0.187	29.344	0.312
Ω_3^ρ	27.783	0.156	31.152	0.140	29.389	0.125
Ω_4^ρ	28.231	0.156	30.631	0.093	29.330	0.125
Ω_5^ρ	27.253	0.312	29.803	0.187	28.502	0.187
Ω_6^ρ	26.972	0.156	30.181	0.234	28.294	0.125
Ω_7^ρ	27.946	0.468	29.955	0.187	28.830	0.312
Ω_8^ρ	27.855	0.156	30.737	0.187	29.079	0.187
Ω_9^ρ	27.622	0.468	30.982	0.281	29.356	0.312

(**a**) DDM (**b**) DDM-CMG (**c**) DDM-NECMG

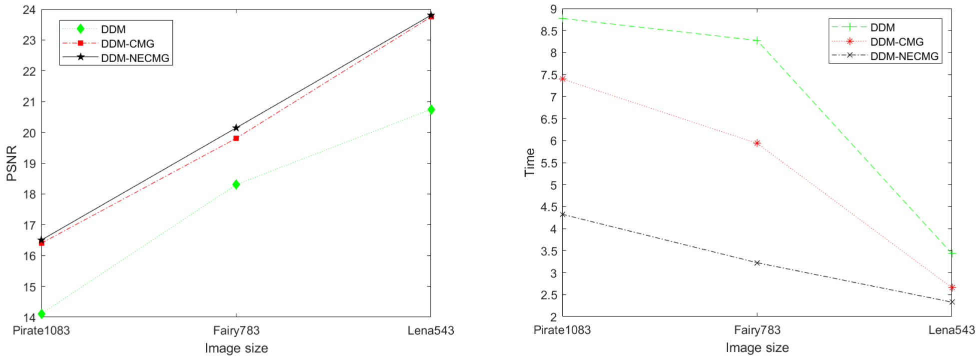
Figure 5. Recovery of Pirate 1083 × 1083.

(**a**) DDM (**b**) DDM-CMG (**c**) DDM-NECMG

Figure 6. Recovery of Fairy 783 × 783.

(**a**) DDM (**b**) DDM-CMG (**c**) DDM-NECMG

Figure 7. Recovery of Lena 543 × 543.



(**a**) DDM (**b**) DDM-CMG

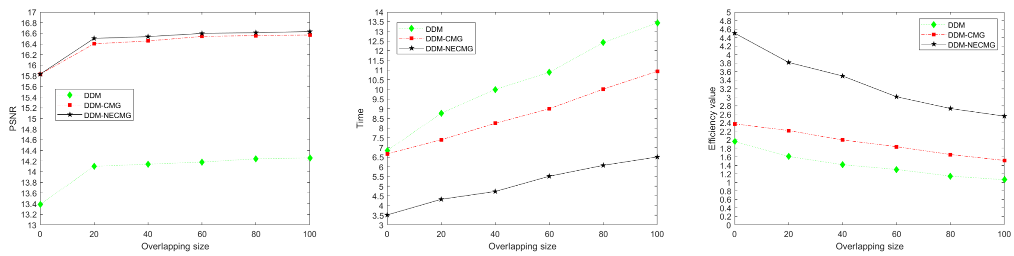
Figure 8. PSNR and CPU times of the three algorithms for different resolution images.

Next, we discuss the influence of the overlap size on image restoration. Taking Pirate 1083 × 1083 as an example, we introduce six different overlapping sizes and analyze them by PSNR and CPU time. The numerical results are shown in Table 4.

From Table 4 and Figure 9, the overlapping domain decomposition method is superior to the non-overlapping domain decomposition method on PSNR, see Figure 9a. However, the CPU time will increase with the increase in overlap size, see Figure 9b. We use the efficiency value (PSNR/CPU time) to represent the performance of the algorithms. By Figure 9c, when $\rho \in (0-20]$, restoration efficiency is suggested.

Table 4. Numerical results for overlapping domains of different sizes

Pirate 1083 × 1083	$\rho = 0$	$\rho = 20$	$\rho = 40$	$\rho = 60$	$\rho = 80$	$\rho = 100$
PSNR (DDM)	13.386	14.101	14.141	14.181	14.242	14.259
PSNR (DDM-CMG)	15.830	16.402	16.456	16.541	16.556	16.567
PSNR (DDM-NECMG)	15.832	16.504	16.537	16.597	16.612	16.632
Time (DDM)	6.839	8.771	9.998	10.882	12.421	13.436
Time (DDM-CMG)	6.671	7.401	8.248	9.001	10.014	10.938
Time (DDM-NECMG)	3.516	4.325	4.729	5.515	6.078	6.512



(**a**) PSNR (**b**) Time (**c**) Efficiency

Figure 9. Recovery of different overlapping domain sizes.

6. Conclusions

In this paper, DDM-CMG and DDM-NECMG are proposed for image restoration problems. The methods partition the image domain into some overlapping subdomains, which can be solved in parallel by using the cascadic multigrid algorithm to reduce the computational and storage requirements. The numerical results show that DDM-CMG and DDM-NECMG can not only improve the computational efficiency but also achieve better denoising and deblurring effects. Compared with the general domain decomposition algorithm, our methods have fast convergence speeds and better recovery qualities.

Author Contributions: Conceptualization, C.L. and Z.C.; methodology, C.L.; software, Z.C.; validation, C.L. and Z.C.; formal analysis, C.L.; resources, C.L.; data curation, C.L. and Z.C.; writing—original draft preparation, Z.C.; writing—review and editing, C.L. and Z.C.; visualization, Z.C.; supervision, C.L.; project administration, C.L.; funding acquisition, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: Natural Science Foundation of China (12161027), Guangxi Natural Science Foundation (2020GXNSFAA159143).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: We would like to thank the Natural Science Foundation of China and Guangxi Natural Science Foundation for supporting this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Adamu, A.; Kumam, P.; Kitkuan, D.; Padcharoen, A. Relaxed modified Tseng algorithm for solving variational inclusion problems in real Banach spaces with applications. *Carpa. J. Math.* **2023**, *39*, 1–26. [Crossref]
2. Adamu, A.; Kitkuan, D.; Padcharoen, A.; Chidume, C.E.; Kumam, P. Inertial viscosity-type iterative method for solving inclusion problems with applications. *Math. Comput. Simulat.* **2022**, *194*, 445–459. [Crossref] [CrossRef]
3. Chidume, C.E.; Adamu, A.; Kumam, P.; Kitkuan, D. Generalized Hybrid Viscosity-Type Forward-Backward Splitting Method with Application to Convex Minimization and Image Restoration Problems. *Numer. Func. Anal. Opt.* **2021**, *42*, 1586–1607. [Crossref] [CrossRef]
4. Morigi, S.; Reichel, L.; Sgallari, F. Cascadic multilevel methods for fast nonsymmetric blur-and noise-removal. *Appl. Numer. Math.* **2010**, *60*, 378–396. [Crossref] [CrossRef]
5. Handlovičová, A.; Mikula, K.; Sgallari, F. Semi-implicit complementary volume scheme for solving level set like equations in image processing and curve evolution. *Numer. Math.* **2003**, *93*, 675–695. [Crossref] [CrossRef]
6. Weickert, J.; Romeny, B.T.H.; Viergever, M.A. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. Image Process.* **1998**, *7*, 398–410. [Crossref] [CrossRef]
7. Morigi, S.; Reichel, L.; Sgallari, F.; Shyshkov, A. Cascadic multiresolution methods for image deblurring. *SIAM J. Imaging Sci.* **2008**, *1*, 51–74. [Crossref] [CrossRef]
8. Marcus, S. Domain decomposition methods. *Appl. Math. Sci. Comput.* **2002**, *224*, 3–29. [Crossref]
9. Firsov, D.; Lui, S. Domain decomposition methods in image denoising using Gaussian curvature. *J. Comput. Appl. Math.* **2006**, *193*, 460–473. [Crossref] [CrossRef]
10. Fornasier, M.; Langer, A.; Schönlieb, C.B. A convergent overlapping domain decomposition method for total variation minimization. *Numer. Math.* **2010**, *116*, 645–685. [Crossref] [CrossRef]
11. Tai, X.C.; Duan, Y.P. Domain decomposition methods with graph cuts algorithms for image segmentation. *Int. J. Numer. Anal. Mod.* **2011**, *8*, 137–155. [Crossref]
12. Duan, Y.P.; Tai, X.C. Domain decomposition methods with graph cuts algorithms for total variation minimization. *Adv. Comput. Math.* **2012**, *36*, 175–199. [Crossref] [CrossRef]
13. Langer, A.; Osher, S.; Schönlieb, C.B. Bregmanized domain decomposition for image restoration. *J. Sci. Comput.* **2013**, *54*, 549–576. [Crossref] [CrossRef]
14. Kamranian, M.; Dehghan, M.; Tatari, M. An image denoising approach based on a meshfree method and the domain decomposition technique. *Eng. Anal. Bound. Elem.* **2014**, *39*, 101–110. [Crossref] [CrossRef]
15. Xu, J.; Chang, H.B.; Qin, J. Domain decomposition method for image deblurring. *J. Comput. Appl. Math.* **2014**, *271*, 401–414. [Crossref] [CrossRef]
16. Chang, H.B.; Zhang, X.Q.; Tai, X.C.; Yang, D.P. Domain decomposition methods for nonlocal total variation image restoration. *J. Sci. Comput.* **2014**, *60*, 79–100. [Crossref] [CrossRef]
17. Chang, H.B.; Tai, X.C.; Yang, D.P. Domain decomposition methods for total variation minimization. *J. Sci. Comput.* **2014**, *1*, 335–349. [Crossref]
18. Kim, Y.; Min, D.; Ham, B.; Sohn, K. Fast domain decomposition for global image smoothing. *IEEE T. Image Process.* **2017**, *26*, 4079–4091. [Crossref] [CrossRef]
19. Chen, R. L. Huang, J.Z.; Tai, X.C. A parallel domain decomposition algorithm for large scale image denoising. *Inverse Probl. Imag.* **2019**, *13*, 1259–1282. [Crossref] [CrossRef]
20. Li, C.L. A new parallel cascadic multigrid method. *Appl. Math. Comput.* **2013**, *219*, 10150–10157. [Crossref] [CrossRef]

21. Ciaramella, G.; Vanzan, T. Substructured two-grid and multi-grid domain decomposition methods. *Numer. Alg.* **2022**, *91*, 413–448. [Crossref] [CrossRef]
22. Liu, Y.W. A new extrapolation cascadic multi-grid method on quasiuniform meshes. *Int. J. Comput. Appl. Math.* **2010**, *5*, 573–584. [Crossref]
23. Morigi, S.; Reichel, L.; Sgallari, F.; Zama, F. Iterative methods for ill-posed problems and semiconvergent sequences. *J. Comput. Appl. Math.* **2006**, *193*, 157–167. [Crossref] [CrossRef]