

A PARALLEL DOMAIN DECOMPOSITION ALGORITHM FOR LARGE SCALE IMAGE DENOISING

RONGLIANG CHEN

Shenzhen Institutes of Advanced Technology
Chinese Academy of Sciences
Shenzhen, Guangdong 518055, China

JIZU HUANG

LSEC, ICMSEC, Academy of Mathematics and Systems Science
Chinese Academy of Sciences
Beijing 100190, China

XIAO-CHUAN CAI*

Department of Computer Science
University of Colorado Boulder
Boulder, CO 80309, USA

(Communicated by Xuecheng Tai)

ABSTRACT. Total variation denoising (TVD) is an effective technique for image denoising, in particular, for recovering blocky, discontinuous images from noisy background. The problem is formulated as an optimization problem in the space of bounded variation functions, and the solution is obtained by solving the associated Euler–Lagrange equation defined on the domain occupied by the entire image. The method offers high quality results, but is computationally expensive for large images, especially for three-dimensional problems. In this paper, we introduce a highly parallel version of the algorithm which formulates the problem as multiple overlapping, but independent, optimization problems, and each is defined on a portion of the image domain. This approach is similar to the overlapping Schwarz type domain decomposition method, but is non-iterative, for solving partial differential equations, and is highly scalable, without using any coarse grids, for parallel computers with a large number of processors. We show by a theory and also by some two- and three-dimensional numerical experiments that the new approach has similar numerical accuracy as the classical TVD approach, but is much more efficient on parallel computers.

1. Introduction. Denoising is usually the first step when processing a digital image since the noise often impact the quality of later phases of the process. Consider a noisy image defined on $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ ($d = 2, 3$):

$$z(\mathbf{x}) = u(\mathbf{x}) + \varepsilon(\mathbf{x}),$$

where $u(\mathbf{x})$ is the ideal image to be restored, $z(\mathbf{x})$ is the observed image, $\varepsilon(\mathbf{x})$ represents the noise to be removed. One of the most successful and popular methods

2010 *Mathematics Subject Classification.* 65N55, 65Y05, 68U10.

Key words and phrases. Image denoising, total variation, overlapping domain decomposition, Newton-Krylov-Schwarz, parallel processing.

* Corresponding author: Xiao-Chuan Cai.

is the Rudin-Osher-Fatemi (ROF) total variation based image denoising method [33]. ROF model can be described as: Find $u \in BV(\Omega)$, such that

$$(1) \quad \min_{u \in BV(\Omega)} \left\{ \alpha \int_{\Omega} |\nabla u| d\Omega + \frac{1}{2} \|u - z\|_{L^2(\Omega)}^2 \right\},$$

where $|\nabla u|$ is the Radon measure, α is a positive parameter, and $BV(\Omega)$ is the space of functions of bounded variation on Ω . The solution of the minimization problem (1) is usually obtained by solving the associated Euler-Lagrange equation which is obtained by the Fréchet differentiation of (1):

$$(2) \quad u - z - \alpha \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) = 0 \quad \text{in } \Omega,$$

with the homogeneous Neumann boundary condition $\frac{\partial u}{\partial \mathbf{n}} = 0$. To avoid the singularity at $|\nabla u| = 0$, a small positive parameter β is usually introduced to the TV functional, and the Euler-Lagrange problem is modified accordingly [33].

There are many techniques available for image denoising, such as the Gaussian smoothing method [4], empirical Wiener filters [10], wavelet thresholding methods [19], nonlocal means methods [6], variation based methods [2], and so on. Because of the capability of preserving sharp edges and boundaries with a high quality recovery, the variation based methods have been widely used with great success [2, 33]. In addition to denoising, the TV method has been successfully applied to other image processing problems, including image deblurring, inpainting, segmentation and cartoon-texture decompositions [5, 9, 11, 30, 38]. In this paper, we focus on the image denoising problem.

Most existing algorithms such as these in [11, 26, 31, 33, 39] for solving (1) target sequential computers. With the recent development of photographic technology, the image resolution is significantly higher. For the denoising of high resolution and large scale images, such as color images, three-dimensional images, the memory requirement and the computing time increase drastically, as a result, parallel computing becomes necessary. There are some recent works for parallel image processing using GPU, multi-core and many-core architectures; see, for example [1, 20, 32, 35]. Single and multilevel domain decomposition (DD) methods were applied to TV minimization problems, and both overlapping and nonoverlapping DD methods were considered. In [23], some non-overlapping DD methods were introduced for the TV model and in [22], some overlapping methods were considered for the same class of problems. A two-level overlapping DD method for the primal total variation minimization was studied in [40], and later they extended the method to the nonlocal TV model in [13]. In [12], a DD method for its dual counterpart was considered and a convergence analysis was provided. In [25], some nonoverlapping DD methods coupled with semi-smooth Newton were studied for the pre-dual TV problem and a convergence analysis was also provided. More recently, a dual decomposition based primal nonoverlapping DD method was introduced in [29] for the TV model. These DD based approaches are more suitable for CPU-based parallelizations, but only limited parallel experiments were reported. In the PhD thesis [27] the author investigated some optimal control and convex programming based nonoverlapping DD methods for smoothed TV and the methods are scalable with up to 144 processor cores. All the above mentioned methods are iterative and require several linear and nonlinear iterations to converge. In this paper, we propose a non-iterative parallel

DD based algorithm applied directly to the optimization problem to solve the large scale image denoising problem on a supercomputer.

To motivate our new algorithm, we make an obvious, but often ignored, observation about a major difference between image denoising problems and problems arising from, for example, solving elliptic partial differential equations (PDE). The difference is about data dependency, which is very important for the purpose of parallelization on large scale parallel computers. Elliptic problems, and many other PDE problems, are globally dependent, in the sense that one can not find the solution on part of the computational domain without knowing the solution on the rest of the domain. On the other hand, image denoising is a local problem, i.e., the noise in part of the domain has nothing to do with the noise on other different parts of the image. The advantage of the noise locality property is not taken into account by the classical ROF approach. In other words, once the optimization problem (1) is formed, the noise becomes a globally connected field. Additional efforts (decomposition by domain and several iterations) will be needed to break up the global field in order to introduce parallelism.

For the purpose for better parallelization, we take advantage of the fact that the noise in different part of the image domain is independent, and form local optimization problems on subdomains, one for each processor of the parallel computer. We note that the algorithm does need a small amount of information from nearby subdomains, therefore, we allow the subdomains to overlap each other. An error analysis is provided in the paper to show that the solution obtained from the new method is essentially the same as the solution of the classical iterative approach based on the global optimization problem, and more importantly, the new method is much faster on parallel computers because there is no communication between processor cores. The algorithm proposed in the paper is aimed to solve minimization problems in the BV space, and the numerical simulations presented in the paper show that the algorithm performs quite well when the observation is in the BV space. However, we need the assumption that the observation $z \in W^{1,1}(\Omega)$ for the theory to work.

The rest of the paper is organized as follows. In Section 2, the new method is introduced and the existence, uniqueness and error analysis of the method are given in Section 3. Some numerical experiments are shown in Section 4 to demonstrate the efficiency and robustness of the algorithm, and to compare the method with the classical ROF approach for two- and three-dimensional problems. Some conclusions and discussions are given in Section 5.

2. A non-iterative overlapping Schwarz method. In this section, we propose a non-iterative overlapping Schwarz method (NiOS) for solving the image denoising problem. The method to be introduced here is different from (1) which is based on an optimization problem defined on the whole domain, but we will show experimentally and theoretically that the difference is small, in terms of the quality of the restored image. The advantage of the new method is that it minimizes the communication cost and is scalable on parallel computers without requiring any coarse grids.

The NIOS method begins with a partition of the image domain into N nonoverlapping subdomains denoted as Ω_0^k ($k = 1, 2, \dots, N$), that satisfy $\Omega = \bigcup_{k=1}^N \Omega_0^k$ and $\Omega_0^i \cap \Omega_0^j = \emptyset$, $i \neq j$. In other words, each pixel belongs to one and only one subdomain. Then we extend each subdomain into a larger subdomain Ω_δ^k ($k = 1, 2, \dots, N$) by including δ layers of pixels (δ is a positive integer) from the

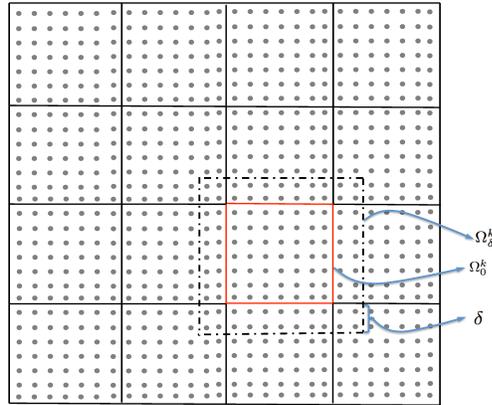


FIGURE 1. An example of the domain partition. Here dots are image pixels and the image domain is partitioned into 16 subdomains. The domain with solid red lines is an example of a nonoverlapping subdomain Ω_0^k and the corresponding overlapping subdomain Ω_δ^k is marked with dashed black lines. The overlapping size δ is the distance between the boundaries of Ω_0^k and Ω_δ^k .

neighboring subdomains; see, for example, Fig. 1. On each subdomain Ω_δ^k , we define a ROF problem as follows:

$$(3) \quad \min_{u^k \in BV(\Omega_\delta^k)} \left\{ \alpha \int_{\Omega_\delta^k} |\nabla u^k| d\Omega_\delta^k + \frac{1}{2} \|u^k - z^k\|_{L^2(\Omega_\delta^k)}^2 \right\},$$

where $z^k = z|_{\Omega_\delta^k}$ is the sub-image with noise on the subdomain Ω_δ^k . The Euler–Lagrange equation corresponding to (3) takes the form

$$(4) \quad \begin{cases} u^k - z^k - \alpha \nabla \cdot \left(\frac{\nabla u^k}{\sqrt{|\nabla u^k|^2 + \beta}} \right) = 0, & \text{in } \Omega_\delta^k, \\ \frac{\partial u^k}{\partial \mathbf{n}} = 0, & \text{on } \partial\Omega_\delta^k, \end{cases}$$

where β is a small positive parameter added to the dominator to avoid the possible singularity when $|\nabla u^k|$ is 0. There are many discretization techniques available for solving (4), such as finite difference method or finite element method. After solving the problem on the overlapping subdomain, we throw away part of the solution and only keep the solution components on the nonoverlapping subdomains Ω_0^k and the denoised image u_h (the discretized solution) is obtained by combining those solutions as follows:

$$(5) \quad u_h = \sum_{k=1}^N E_0^k R_0^k u_h^k,$$

where R_0^k is a restriction operator from the overlapping subdomain Ω_δ^k to the nonoverlapping subdomain Ω_0^k , that is, for a vector u_h^k defined on Ω_δ^k , $R_0^k u_h^k$ takes components belonging only to Ω_0^k . E_0^k is an extension operator from the nonoverlapping subdomain Ω_0^k to the whole domain Ω which is defined as follows. Let m be the

total number of pixels in Ω and m^k the total number of pixels in Ω_δ^k . Then the extension operator E_0^k is a $m \times m^k$ matrix, whose elements $(E_0^k)_{l_1, l_2}$ are 1 if $1 \leq l_1 \leq m$ and $1 \leq l_2 \leq m^k$ correspond to a pixel in Ω_0^k or 0 otherwise. In other words, we extend the local vector u_h^k to a global vector by adding zeros $(0, \dots, 0, R_0^k u_h^k, 0, \dots, 0)$. Fig. 2 gives an example for the $N = 4$ case. The framework of the NiOS algorithm is given in Algorithm 1.

Algorithm 1 : NiOS

- Step 1. Partition the original image into N nonoverlapping sub-images Ω_0^k ($k = 1, 2, \dots, N$) and then extend each sub-image to include δ layers of pixels surrounding the sub-image Ω_0^k to form overlapping subdomains.
- Step 2. Define the ROF model (3) on the overlapping subdomains Ω_δ^k ($k = 1, 2, \dots, N$), derive the Euler–Lagrange equation (4), and solve equation (4) in parallel to obtain u_h^k .
- Step 3. Combine the sub-problem solutions u_h^k ($k = 1, 2, \dots, N$) into a global solution u_h using (5).
-

We remark that Algorithm 1 is similar to the overlapping domain decomposition method that decomposes the computational domain into overlapping subdomains and then restricts the original partial differential equation to each of the subdomains, except here we apply the overlapping decomposition idea directly to the optimization problem. The other difference is that DD usually requires several iterations to obtain the solution. However, numerical experiments in this paper indicate that the NiOS algorithm can obtain a sufficiently accurate solution without any iteration. To understand this, an error analysis is provided in the next section to investigate the difference between the solutions obtained by NiOS and the classical ROF method based on solving the global optimization problem. Note also that traditional DD requires coarse grids in order to be scalable, but NiOS is scalable without any coarse grids. It is important to note that the increase of the overlapping size is helpful to reduce the difference between NiOS and the classical ROF but it does not improve much of the quality of the image measured by the peak signal-to-noise ratio. To balance the accuracy and efficiency of the NiOS algorithm, a relatively small overlapping size should be chosen according to our experimental studies.

3. Existence, uniqueness, and error analysis. In this section, we first prove the existence and uniqueness of the solution for the Euler–Lagrange equation. Then an auxiliary one-dimensional problem is studied. Finally we provide an error analysis which shows that the solution obtained by NiOS is close to the solution obtained by the method based on the global optimization approach.

We first consider the existence and uniqueness of the solution of the Euler–Lagrange equation

$$(6) \quad u - z - \alpha \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) = 0 \quad \text{in } \Omega,$$

with a homogeneous Neumann boundary condition

$$\frac{\partial u}{\partial \mathbf{n}} = 0.$$

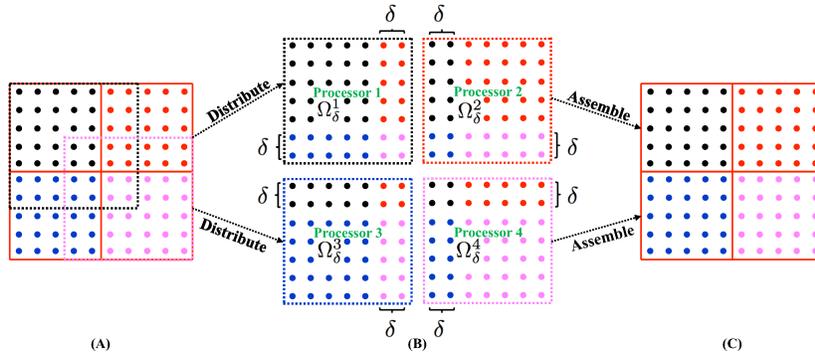


FIGURE 2. Procedure of NiOS. **Step 1:** Overlapping domain decomposition (A); **Step 2:** Distribute overlapping subdomain problems to 4 processors and solve these four overlapping sub-problems (B); **Step 3:** Combine images from non-overlapping subdomains (C). Here the points with different colors are the image pixels distributed to different processor cores.

We are mainly interested in the case that Ω is a bounded domain with Lipschitz boundary. By denoting the vector function $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^d$ as

$$\Phi(\mathbf{x}) = \frac{\mathbf{x}}{\sqrt{|\mathbf{x}|^2 + \beta}},$$

we can rewrite (6) as a capillarity type equation:

$$(7) \quad \nabla \cdot \Phi(\nabla u(\mathbf{x})) = \varphi(\mathbf{x}, u) \quad \text{in } \Omega.$$

When the boundary condition takes the form $\Phi(\nabla u) \cdot \mathbf{n} = \cos \theta$, the problem was carefully studied in [24, 36, 37]. In these papers, $\varphi(\mathbf{x}, v)$ is a given function and assumed to have Hölder continuous partial derivatives with respect to $(\mathbf{x}, v) \in \bar{\Omega} \times \mathbb{R}$ and satisfy

$$\inf_{\mathbf{x} \in \bar{\Omega}, v \in \mathbb{R}} \frac{\partial \varphi(\mathbf{x}, v)}{\partial v} > 0,$$

where $\bar{\Omega}$ represents the closure of Ω . Ural'tseva [37] shows that (7) has a $C^2(\bar{\Omega})$ solution if $\theta = \text{constant} \in (0, \pi)$ and Ω is a convex domain of class $C^{2,\gamma}$. Using a different method, Spruck [36] obtains a similar result in $\Omega \in \mathbb{R}^2$ with C^4 smooth boundary without the restriction of convexity. Moreover, the angle θ is allowed to be non-constant, and the cases $\theta = 0, \pi$ are also discussed. Similar results are reported in [24].

In this paper, we aim to estimate the error of NiOS on a bounded domain with Lipschitz boundary. Note that the noisy image z is usually considered as a function in $L^2(\Omega)$ and in such a situation $\varphi(\mathbf{x}, v)$ may not have Hölder continuous partial derivatives with respect to $\mathbf{x} \in \bar{\Omega}$. In such situation, the strong regularity results of [24, 36, 37] do not hold for the Euler–Lagrange equation, we therefore switch to a weak form of the problem. We first prove that the Euler–Lagrange equation has a unique weak solution in $W^{1,1}(\Omega) \cap L^2(\Omega)$. The weak form of (7) is: Find

$u \in W^{1,1}(\Omega)$ such that

$$\int_{\Omega} (u - z)\phi d\Omega + \int_{\Omega} \alpha \nabla \phi \cdot \Phi(\nabla u) d\Omega = 0,$$

holds for any $\phi \in W^{1,1}(\Omega)$.

Lemma 3.1. *The following inequality holds for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ with $d = 1, 2$ or 3 ,*

$$(\Phi(\mathbf{x}) - \Phi(\mathbf{y})) \cdot (\mathbf{x} - \mathbf{y}) \geq 0.$$

Proof.

$$\begin{aligned} & (\Phi(\mathbf{x}) - \Phi(\mathbf{y})) \cdot (\mathbf{x} - \mathbf{y}) \\ &= \frac{|\mathbf{x}|^2}{\sqrt{|\mathbf{x}|^2 + \beta}} + \frac{|\mathbf{y}|^2}{\sqrt{|\mathbf{y}|^2 + \beta}} - \frac{\mathbf{x} \cdot \mathbf{y}}{\sqrt{|\mathbf{x}|^2 + \beta}} - \frac{\mathbf{x} \cdot \mathbf{y}}{\sqrt{|\mathbf{y}|^2 + \beta}} \\ &\geq \frac{|\mathbf{x}|^2}{\sqrt{|\mathbf{x}|^2 + \beta}} + \frac{|\mathbf{y}|^2}{\sqrt{|\mathbf{y}|^2 + \beta}} - \frac{|\mathbf{x}||\mathbf{y}|}{\sqrt{|\mathbf{x}|^2 + \beta}} - \frac{|\mathbf{x}||\mathbf{y}|}{\sqrt{|\mathbf{y}|^2 + \beta}} \\ &= (|\mathbf{x}| - |\mathbf{y}|) \left(\frac{|\mathbf{x}|}{\sqrt{|\mathbf{x}|^2 + \beta}} - \frac{|\mathbf{y}|}{\sqrt{|\mathbf{y}|^2 + \beta}} \right) \geq 0. \end{aligned}$$

The last inequality holds due to the monotonicity of the function $|\Phi|$. □

Theorem 3.1. *Let Ω be a bounded domain with Lipschitz boundary. Suppose $z \in W^{1,1}(\Omega) \cap L^2(\Omega)$. α, β are two positive constants. Then the Euler–Lagrange equation has a unique weak solution $u \in W^{1,1}(\Omega) \cap L^2(\Omega)$ satisfying $\|u\|_{L^2(\Omega)} \leq \|z\|_{L^2(\Omega)}$ and $\|\nabla u\|_{L^1(\Omega)} \leq C + \sqrt{1 + \beta} \|\nabla z\|_{L^1(\Omega)}$, where C is a positive constant independent of α and β .*

Proof. According to the variation theory, the solution of the Euler–Lagrange equation is also the solution of the following minimization problem

$$(8) \quad \mathcal{L}(u) = \min_{v \in V(\Omega)} \mathcal{L}(v),$$

where the function $\mathcal{L}(v)$ is defined as $\mathcal{L}(v) := \alpha \int_{\Omega} (\beta + |\nabla v|^2)^{1/2} d\Omega + \frac{1}{2} \|v - z\|_{L^2(\Omega)}^2$ and the space $V(\Omega)$ is defined as $V(\Omega) := \{v \in W^{1,1}(\Omega) | \nabla v \cdot \mathbf{n} = 0 \text{ on } \partial\Omega\}$. It is easy to check that $\mathcal{L}(v) \geq \alpha \|\nabla v\|_{L^1(\Omega)}$ and $\mathcal{L}(v)$ is convex. Thus, similar to [21] (Theorem 2, page 448), we obtain that there exists at least one solution $u \in V(\Omega)$ such that $\mathcal{L}(u) = \min_{v \in V(\Omega)} \mathcal{L}(v)$ by replacing the Dirichlet boundary condition with the

homogeneous Neumann boundary condition, which means that the Euler–Lagrange equation exists at least one solution in $V(\Omega)$.

Since the function $\mathcal{L}(u)$ is strictly convex, the minimization problem exists only one solution, which implies the uniqueness of the weak solution of the Euler–Lagrange equation.

Multiplying the Euler–Lagrange equation with u and taking the integral, we obtain

$$\int_{\Omega} u^2 d\Omega + \int_{\Omega} \alpha \nabla u \cdot \Phi(\nabla u) d\Omega = \int_{\Omega} uz d\Omega \leq \frac{1}{2} (\|u\|_{L^2(\Omega)}^2 + \|z\|_{L^2(\Omega)}^2).$$

Since the second term in the left-hand side is nonnegative, this implies $\|u\|_{L^2(\Omega)} \leq \|z\|_{L^2(\Omega)}$. Because $z \in W^{1,1}(\Omega)$, there is a positive constant C such that $\|\nabla z\|_{L^1(\Omega)} \leq$

C. Let $\tilde{\Omega}$ be a subdomain of Ω defined as $\tilde{\Omega} = \{\mathbf{x} \in \Omega \mid |\nabla u| \geq 1\}$. Then we have

$$\frac{\alpha}{\sqrt{1+\beta}} \|\nabla u\|_{L^1(\tilde{\Omega})} \leq \int_{\tilde{\Omega}} \alpha \nabla u \cdot \Phi(\nabla u) d\tilde{\Omega} \leq \int_{\Omega} \alpha \nabla z \cdot \Phi(\nabla u) d\Omega \leq \alpha \|\nabla z\|_{L^1(\Omega)},$$

which shows that $\|\nabla u\|_{L^1(\tilde{\Omega})} \leq \sqrt{1+\beta} \|\nabla z\|_{L^1(\Omega)}$. Subsequently, we conclude that $\|\nabla u\|_{L^1(\Omega)} \leq C + \sqrt{1+\beta} \|\nabla z\|_{L^1(\Omega)}$, where C is a constant independent of α and β . The proof is complete. \square

In the rest of this section, we assume $\beta < 1$ and α is a given positive constant. We introduce an auxiliary one-dimensional problem as follows, and later we will consider two- and three-dimensional problems as tensor products of one-dimensional problems

$$(9) \quad \begin{cases} v(y) - \alpha D\Psi(Dv(y)) = 0 & y \in (a, b), \\ Dv|_{y=a} = \nu_1, \quad Dv|_{y=b} = \nu_2. \end{cases}$$

Here D represents the derivative and the scalar function $\Psi : \mathbb{R} \rightarrow \mathbb{R}$ is defined as $\Psi(y) = \frac{y}{\sqrt{y^2+\beta}}$.

Lemma 3.2. *Assume $\nu_1\nu_2 = 0$. The auxiliary problem (9) has a unique solution that satisfies*

$$(10) \quad \begin{aligned} \text{osc}_{y \in [a,b]} v(y) &\leq C\beta^{1/4}, \\ \max_{y \in [a,b]} |v(y)| &\leq C(\alpha + \beta^{1/4}), \\ |Dv(y)| &\leq C \quad \text{in } [a, b], \\ |Dv(y)| &\leq C\beta^{1/4} \quad \text{in } [a - C\beta^{1/2} \log \beta, b + C\beta^{1/2} \log \beta]. \end{aligned}$$

Here C is a positive constant independent of α and β . osc refers to the oscillation defined as $\text{osc}_{y \in [a,b]} v(y) = \sup_{y \in [a,b]} v(y) - \inf_{y \in [a,b]} v(y)$.

Proof. The uniqueness of the solution is easily obtained from Theorem 3.1. Let $\sigma = \sqrt{\frac{\alpha}{\beta^{1/2}}}$, $\bar{y} = \sigma y$, and $\bar{v} = v/(\sigma\beta^{1/2})$. Then, \bar{v} is a solution of a capillarity type equation as follows

$$\bar{v} - D_{\bar{y}} \left(\frac{D_{\bar{y}} \bar{v}}{\sqrt{(D_{\bar{y}} \bar{v})^2 + 1}} \right) = 0 \quad \text{in } (\sigma a, \sigma b).$$

By using Theorem 3.3 in [36], we have $\bar{v} \in C^3[a, b]$ and $v \in C^3[a, b]$. Next, we prove $|Dv| \leq C$ with C independent of α and β . Let $\mathcal{Y} = \{y \in [a, b] \mid |Dv(y)| = \max_{z \in [a,b]} |Dv(z)|\}$ and $y_0 = \min \mathcal{Y}$. If $y_0 \in (a, b)$, we obtain $D^2v(y_0) = 0$. According to

(9), we have $v(y_0) = \alpha\beta \frac{D^2v(y_0)}{(\sqrt{(Dv(y_0))^2 + \beta})^3} = 0$. If $Dv(y_0) > 0$, it follows $D^2v(y) > 0$ and $v(y) < 0$ for $y \in (y_0 - \varepsilon, y_0)$ with small enough ε , which contradicts with $v(y_0) = \alpha\beta \frac{D^2v(y_0)}{(\sqrt{(Dv(y_0))^2 + \beta})^3}$. Similarly, for $Dv(y_0) < 0$, we have $D^2\bar{v}(y) < 0$ and $v(y) > 0$ for $y \in (y_0 - \varepsilon, y_0)$ with small enough ε , which also contradicts with $v(y_0) = \alpha\beta \frac{D^2v(y_0)}{(\sqrt{(Dv(y_0))^2 + \beta})^3}$. Thus, we have $y_0 = a$ or $y_0 = b$ and $|Dv| \leq \max\{|\nu_1|, |\nu_2|\} \leq C$, where C is a constant independent of α and β .

If $\nu_1 = \nu_2 = 0$, then the solution of (9) is $v = 0$ and satisfies (10), which completes the proof in this situation. Next, we consider the case with $\nu_1 = 0$ and $\nu_2 > 0$. The proof of the other case is similar. Let us assume that the minimum value

$v(\tilde{y}_0) = \min_{y \in [a,b]} v(y) < 0$. Due to $\nu_1 = 0, \nu_2 > 0$ and the minimum condition, we get $D^2v(\tilde{y}_0) \geq 0$ and $Dv(\tilde{y}_0) = 0$. However, $D^2v(\tilde{y}_0) = v(\tilde{y}_0) \frac{[(Dv(\tilde{y}_0))^2 + \beta]^{3/2}}{\alpha\beta} < 0$, which shows the assumption is not true. Thus $v(y) \geq 0$ and $D^2v(y) \geq 0$ hold for all $y \in [a, b]$, which implies $Dv(y)$ is monotonically increasing in $[a, b]$. Due to $\nu_1 = 0$, we have $Dv(y) \geq 0$ and $v(y)$ is a monotonically increasing function in $[a, b]$. If $\nu_2 \leq \beta^{1/4}$, it is clear that $\text{osc}_{y \in [a,b]} v(y) \leq C\beta^{1/4}$ and $|Dv(y)| \leq C\beta^{1/4}$ hold for all

$y \in [a, b]$. Since $\int_a^b v dy = \alpha\Phi(\nu_2)$, we obtain

$$v(b) = \max_{y \in [a,b]} |v(y)| \leq C(\alpha + \beta^{1/4}),$$

which implies that the estimates (10) hold.

Next, we assume $\nu_2 > \beta^{1/4}$. Since $v(y)$ is monotonically increasing in $[a, b]$, we have $v(a) = \min_{y \in [a,b]} v(y) \geq 0$. Let ζ_1 in $[a, b]$ such that $Dv(\zeta_1) = \beta^{1/2}$. Then we have

$$\beta^{1/2} = \int_a^{\zeta_1} D^2v dy = \int_a^{\zeta_1} \frac{[(Dv)^2 + \beta]^{3/2}}{\alpha\beta} v dy \leq 2^{3/2} \int_a^{\zeta_1} \frac{\beta^{1/2} v(\zeta_1)}{\alpha} dy,$$

which implies $(\zeta_1 - a)v(\zeta_1) \geq C\alpha$ and $v(\zeta_1) \geq C\alpha$.

We only consider the case with $\nu_2 > \beta^{1/6}$. The other cases are similar. Since $Dv(y)$ is monotonically increasing in $[a, b]$, we can choose a sequence ξ_n such that $Dv(\xi_n) = \beta^{\gamma_n}$, for $n = 1, 2, \dots$. Here $\gamma_1 = 1/6$ and $\gamma_n = 1/6 + \gamma_{n-1}/3$. It is clear $1/4 = \lim_{n \rightarrow \infty} \gamma_n, \gamma_n \leq 1/4$, and $\lim_{n \rightarrow \infty} \xi_n = \xi_0$ with $Dv(\xi_0) = \beta^{1/4}$. Since $\xi_i > \zeta_1$ and $v(\xi_i)/\alpha \geq C$, we have

$$(11) \quad \nu_2 - \beta^{1/6} = Dv(b) - Dv(\xi_1) = \int_{\xi_1}^b D^2v dy = \int_{\xi_1}^b \frac{[(Dv)^2 + \beta]^{3/2}}{\alpha\beta} v dy \geq \int_{\xi_1}^b \frac{\beta^{-1/2} v}{\alpha} dy,$$

which implies $b - \xi_1 \leq C\beta^{1/2}\alpha$. Similarly, we get

$$(12) \quad \begin{aligned} 1 - \beta^{\gamma_n - \gamma_{n-1}} &= \beta^{-\gamma_{n-1}} \int_{\xi_n}^{\xi_{n-1}} D^2v dy = \beta^{-\gamma_{n-1}} \int_{\xi_n}^{\xi_{n-1}} \frac{[(Dv)^2 + \beta]^{3/2}}{\alpha\beta} v dy \\ &\geq \beta^{-\gamma_{n-1}} \int_{\xi_n}^{\xi_{n-1}} \frac{(\beta^{2\gamma_n} + \beta)^{3/2}}{\alpha\beta} v dy \geq \int_{\xi_n}^{\xi_{n-1}} \frac{\beta^{-1/2} v}{\alpha} dy. \end{aligned}$$

Together with (11) and (12), we have

$$\nu_2 - \beta^{\gamma_1} + \sum_{i=2}^n (1 - \beta^{\gamma_n - \gamma_{n-1}}) \geq \int_{\xi_n}^b \frac{\beta^{-1/2} v}{\alpha} dy.$$

Since $-y \log \beta \geq (1 - \beta^y)$ holds for any small positive y , we get

$$C \log \beta \geq \nu_2 - \beta^{\gamma_1} - (\gamma_1 - \gamma_n) \log \beta \geq \int_{\xi_n}^b \frac{\beta^{-1/2} v}{\alpha} dy.$$

Then we obtain $b - \xi_n \leq -C\beta^{1/2} \log \beta$. Let $n \rightarrow \infty$, we have $b - \xi_0 \leq -C\beta^{1/2} \log \beta$, which shows the fourth inequality of (10) is true. Since $Dv(y) \geq 0$, we have

$$\text{osc}_{y \in [a,b]} v(y) = \int_a^b Dv(y) dy \leq C\beta^{1/4},$$

which implies the first inequality of (10) holds. Due to $\int_a^b v dy = \alpha\Phi(\nu_2)$, it follows that $\max_{y \in [a,b]} |v(y)| \leq C(v(a) + \beta^{1/4}) \leq C(\alpha + \beta^{1/4})$, which completes the proof. \square

Now let us consider the error estimates for the two- and three-dimensional problems based on the above Lemmas and Theorem. Assume u is the solution of (6) on domain Ω and u^k is the solution of the sub-problem on Ω_δ^k . Here $\Omega_\delta^k = ((x_1^s, x_1^u) \times \cdots \times (x_d^s, x_d^u)) \cap \Omega$. From Theorem 3.1, we get the uniqueness of the two solutions. For a fixed k , it is clear that u satisfies

$$u - z - \alpha \nabla \cdot \Phi(\nabla u) = 0 \quad \text{in } \Omega_\delta^k,$$

with boundary condition $\Phi(\nabla u) \cdot \mathbf{n} = \nu(\mathbf{x})$ for \mathbf{x} on $\partial\Omega_\delta^k$. It is clear that $|\nu(\mathbf{x})| \leq 1$ and $\nu(\mathbf{x}) = 0$ for \mathbf{x} on $\partial\Omega$. The following theorem provides an estimate of the distance between u and u^k in $L^2(\Omega_\delta^k)$.

Theorem 3.2. *Let Ω be a bounded domain with Lipschitz boundary and $\Omega_\delta^k = ((x_1^s, x_1^u) \times \cdots \times (x_d^s, x_d^u)) \cap \Omega$. Suppose $z \in W^{1,1}(\Omega) \cap L^2(\Omega)$. Assume u is the solution of (6) on Ω and u^k is the solution of (4) on Ω_δ^k . Then we have*

$$\|u - u^k\|_{L^2(\Omega_\delta^k)}^2 \leq C(\alpha + \beta^{1/4}).$$

Here C is a constant which is dependent on $\|u\|_{W^{1,1}(\Omega)}$ but independent of α and β .

Proof. We obtain the uniquenesses of u and u^k from Theorem 3.1. For $i = 1, 2, \dots, d$, let w_i^s and w_i^u be the solutions of the following equations

$$(13) \quad \begin{cases} w_i^s(x_i) - \alpha D(\Psi(Dw_i^s(x_i))) = 0 & \text{in } (x_i^s, x_i^u), \\ Dw_i^s(x_i)|_{x_i=x_i^s} = -1, \quad Dw_i^s(x_i)|_{x_i=x_i^u} = 0, \end{cases}$$

and

$$(14) \quad \begin{cases} w_i^u(x_i) - \alpha D(\Psi(Dw_i^u(x_i))) = 0 & \text{in } (x_i^s, x_i^u), \\ Dw_i^u(x_i)|_{x_i=x_i^s} = 0, \quad Dw_i^u(x_i)|_{x_i=x_i^u} = 1. \end{cases}$$

Case A. $\Omega_\delta^k = (x_1^s, x_1^u) \times \cdots \times (x_d^s, x_d^u) \subset \Omega$. In this case, Ω_δ^k is a rectangular or hexahedral subdomain of Ω . With the definition of u, u^k, w_i^s, w_i^u , we have $u - u^k - \sum_{i=1}^d \tilde{\nu}_i^s w_i^s - \sum_{i=1}^d \tilde{\nu}_i^u w_i^u$ satisfies

$$\begin{cases} u - u^k - \sum_{i=1}^d \tilde{\nu}_i^s w_i^s - \sum_{i=1}^d \tilde{\nu}_i^u w_i^u - \alpha \nabla \cdot \Phi(\nabla u) + \alpha \nabla \cdot \Phi(\nabla u^k) \\ + \sum_{i=1}^d \alpha \tilde{\nu}_i^s \partial_{x_i} (\Psi(\partial_{x_i} w_i^s)) + \sum_{i=1}^d \alpha \tilde{\nu}_i^u \partial_{x_i} (\Psi(\partial_{x_i} w_i^u)) = 0 \quad \text{in } \Omega_\delta^k, \end{cases}$$

where $\tilde{\nu}_i^s = -\sqrt{(1 + \beta)\Phi(\nabla u(\mathbf{x}^s)) \cdot \mathbf{n}}$, and $\tilde{\nu}_i^u = \sqrt{(1 + \beta)\Phi(\nabla u(\mathbf{x}^u)) \cdot \mathbf{n}}$. Here $\mathbf{x}^s = (x_1, \dots, x_{i-1}, x_i^s, x_{i+1}, \dots, x_d)$ and $\mathbf{x}^u = (x_1, \dots, x_{i-1}, x_i^u, x_{i+1}, \dots, x_d)$ are

two points on $\partial\Omega_\delta^k$. By multiplying the above equation by $u - u^k$ and taking the integral, we get

$$\begin{aligned}
 (15) \quad & \|u - u^k\|_{L^2(\Omega_\delta^k)}^2 + \alpha \int_{\Omega_\delta^k} \left(\Phi(\nabla u) - \Phi(\nabla u^k) \right) \nabla(u - u^k) d\Omega_\delta^k \\
 &= \sum_{i=1}^d \alpha \int_{\Omega_\delta^k} \tilde{\nu}_i^s \Psi(\partial_{x_i} w_i^s) \partial_{x_i}(u - u^k) d\Omega_\delta^k + \sum_{i=1}^d \alpha \int_{\Omega_\delta^k} \tilde{\nu}_i^u \Psi(\partial_{x_i} w_i^u) \partial_{x_i}(u - u^k) d\Omega_\delta^k \\
 &+ \int_{\Omega_\delta^k} \sum_{i=1}^d \tilde{\nu}_i^s w_i^s (u - u^k) d\Omega_\delta^k + \int_{\Omega_\delta^k} \sum_{i=1}^d \tilde{\nu}_i^u w_i^u (u - u^k) d\Omega_\delta^k.
 \end{aligned}$$

The first two terms in the right-hand side of (15) satisfy

$$\begin{aligned}
 & \sum_{i=1}^d \alpha \int_{\Omega_\delta^k} \tilde{\nu}_i^s \Psi(\partial_{x_i} w_i^s) \partial_{x_i}(u - u^k) d\Omega_\delta^k + \sum_{i=1}^d \alpha \int_{\Omega_\delta^k} \tilde{\nu}_i^u \Psi(\partial_{x_i} w_i^u) \partial_{x_i}(u - u^k) d\Omega_\delta^k \\
 &= \sum_{i=1}^d \alpha \int_{\Omega_\delta^k} \partial_{x_i}(u - u^k) d\Omega_\delta^k \\
 &\quad \left(\tilde{\nu}_i^u \int_{x_i^s}^{x_i} \partial_{x_i}(\Psi(\partial_{x_i} w_i^u)) dx_i - \tilde{\nu}_i^s \int_{x_i}^{x_i^u} \partial_{x_i}(\Psi(\partial_{x_i} w_i^s)) dx_i \right) \\
 &= \sum_{i=1}^d \int_{\Omega_\delta^k} \partial_{x_i}(u - u^k) d\Omega_\delta^k \left(\tilde{\nu}_i^u \int_{x_i^s}^{x_i} w_i^u dx_i - \tilde{\nu}_i^s \int_{x_i}^{x_i^u} w_i^s dx_i \right).
 \end{aligned}$$

Thanks to Theorem 3.1 and Lemma 3.2, we obtain

$$\begin{aligned}
 & \left| \sum_{i=1}^d \alpha \int_{\Omega_\delta^k} \tilde{\nu}_i^s \Psi(\partial_{x_i} w_i^s) \partial_{x_i}(u - u^k) d\Omega_\delta^k + \sum_{i=1}^d \alpha \int_{\Omega_\delta^k} \tilde{\nu}_i^u \Psi(\partial_{x_i} w_i^u) \partial_{x_i}(u - u^k) d\Omega_\delta^k \right| \\
 & \leq C(\alpha + \beta^{1/4}).
 \end{aligned}$$

The last two terms in the right-hand side of (15) satisfy

$$\begin{aligned}
 & \left| \int_{\Omega_\delta^k} \sum_{i=1}^d \tilde{\nu}_i^s w_i^s (u - u^k) d\Omega_\delta^k + \int_{\Omega_\delta^k} \sum_{i=1}^d \tilde{\nu}_i^u w_i^u (u - u^k) d\Omega_\delta^k \right| \\
 & \leq C(\alpha + \beta^{1/4}) \|u - u^k\|_{L^2(\Omega_\delta^k)}.
 \end{aligned}$$

Since the second term in the left-hand side of (15) is nonnegative, we have

$$\|u - u^k\|_{L^2(\Omega_\delta^k)}^2 \leq C(\alpha + \beta^{1/4}),$$

which completes the proof in this situation.

Case B. $\Omega_\delta^k \subsetneq (x_1^s, x_1^u) \times \dots \times (x_d^s, x_d^u)$. In this case, $\partial\Omega_\delta^k \cap \partial\Omega \neq \emptyset$. For a point $\mathbf{x} \in \Omega_\delta^k$, let us define $\rho_i(\mathbf{x}) = x_i^s + \frac{(x_i - \tilde{x}_i^s)(x_i^u - x_i^s)}{(\tilde{x}_i^u - \tilde{x}_i^s)}$, where $\tilde{\mathbf{x}}^s = (x_1, \dots, x_{i-1}, \tilde{x}_i^s, x_{i+1}, \dots, x_d)$ and $\tilde{\mathbf{x}}^u = (x_1, \dots, x_{i-1}, \tilde{x}_i^u, x_{i+1}, \dots, x_d)$ are two points on $\partial\Omega_\delta^k$. We can choose the two points $\tilde{\mathbf{x}}^s$ and $\tilde{\mathbf{x}}^u$ such that the line between them belongs to Ω_δ^k and $\tilde{x}_i^u > \tilde{x}_i^s$.

Then, we have $u - u^k - \sum_{i=1}^d \tilde{\nu}_i^s w_i^s(\rho_i) - \sum_{i=1}^d \tilde{\nu}_i^u w_i^u(\rho_i)$ satisfies

$$u - u^k - \sum_{i=1}^d \tilde{\nu}_i^s w_i^s(\rho_i) - \sum_{i=1}^d \tilde{\nu}_i^u w_i^u(\rho_i) - \alpha \nabla \cdot \Phi(\nabla u) + \alpha \nabla \cdot \Phi(\nabla u^k) + \sum_{i=1}^d \alpha \tilde{\nu}_i^s \partial_{\rho_i} (\Psi(\partial_{\rho_i} w_i^s(\rho_i))) + \sum_{i=1}^d \alpha \tilde{\nu}_i^u \partial_{\rho_i} (\Psi(\partial_{\rho_i} w_i^u(\rho_i))) = 0 \quad \text{in } \Omega_\delta^k,$$

where $\tilde{\nu}_i^s = -\sqrt{(1 + \beta)}\Phi(\nabla u(\tilde{\mathbf{x}}^s)) \cdot \mathbf{n}$ and $\tilde{\nu}_i^u = \sqrt{(1 + \beta)}\Phi(\nabla u(\tilde{\mathbf{x}}^u)) \cdot \mathbf{n}$, respectively. Here w_i^s are w_i^u are given in (13) and (14), respectively. It is important to note that $\tilde{\nu}_i^u$ and $\tilde{\nu}_i^s$ are both equal to zero while $\tilde{\mathbf{x}}^u$ and $\tilde{\mathbf{x}}^s$ are on $\partial\Omega$. For $\tilde{\mathbf{x}}^u$ or $\tilde{\mathbf{x}}^s$ on $\partial\Omega_\delta^k \setminus \partial\Omega$, we have that $\tilde{\nu}_i^u$ or $\tilde{\nu}_i^s$ is a function independent of x_i . Similar to the case A, multiplying the above equation by $u - u^k$ and taking the integral give that (16)

$$\begin{aligned} & \|u - u^k\|_{L^2(\Omega_\delta^k)}^2 + \alpha \int_{\Omega_\delta^k} (\Phi(\nabla u) - \Phi(\nabla u^k)) \nabla(u - u^k) d\Omega_\delta^k \\ &= \sum_{i=1}^d \alpha \frac{(\tilde{x}_i^u - \tilde{x}_i^s)}{(x_i^u - x_i^s)} \int_{\Omega_\delta^k} \left\{ \tilde{\nu}_i^s \Psi(\partial_{\rho_i} w_i^s(\rho_i)) + \tilde{\nu}_i^u \Psi(\partial_{\rho_i} w_i^u(\rho_i)) \right\} \partial_{x_i}(u - u^k) d\Omega_\delta^k \\ &+ \sum_{i=1}^d \int_{\Omega_\delta^k} \left\{ \tilde{\nu}_i^s w_i^s(\rho_i) + \tilde{\nu}_i^u w_i^u(\rho_i) \right\} (u - u^k) d\Omega_\delta^k. \end{aligned}$$

The first summation in the right-hand side of (16) satisfies

$$\begin{aligned} & \left| \sum_{i=1}^d \alpha \frac{(\tilde{x}_i^u - \tilde{x}_i^s)}{(x_i^u - x_i^s)} \int_{\Omega_\delta^k} \left\{ \tilde{\nu}_i^s \Psi(\partial_{\rho_i} w_i^s(\rho_i)) + \tilde{\nu}_i^u \Psi(\partial_{\rho_i} w_i^u(\rho_i)) \right\} \partial_{x_i}(u - u^k) d\Omega_\delta^k \right| \\ & \leq \left| \sum_{i=1}^d \alpha \int_{\Omega_\delta^k} \partial_{x_i}(u - u^k) d\Omega_\delta^k \left(\tilde{\nu}_i^u \int_{x_i^s}^{x_i^u} \partial_{\rho_i} (\Psi(\partial_{\rho_i} w_i^u(\rho_i))) d\rho_i \right. \right. \\ & \qquad \qquad \qquad \left. \left. - \tilde{\nu}_i^s \int_{x_i}^{x_i^u} \partial_{\rho_i} (\Psi(\partial_{\rho_i} w_i^s(\rho_i))) d\rho_i \right) \right| \\ & = \left| \sum_{i=1}^d \int_{\Omega_\delta^k} \partial_{x_i}(u - u^k) d\Omega_\delta^k \left(\tilde{\nu}_i^u \int_{x_i^s}^{x_i^u} w_i^u(\rho_i) d\rho_i - \tilde{\nu}_i^s \int_{x_i}^{x_i^u} w_i^s(\rho_i) d\rho_i \right) \right|. \end{aligned}$$

Thanks to Theorem 3.1 and Lemma 3.2, we obtain

$$\begin{aligned} & \left| \sum_{i=1}^d \alpha \frac{(\tilde{x}_i^u - \tilde{x}_i^s)}{(x_i^u - x_i^s)} \int_{\Omega_\delta^k} \left\{ \tilde{\nu}_i^s \Psi(\partial_{\rho_i} w_i^s(\rho_i)) + \tilde{\nu}_i^u \Psi(\partial_{\rho_i} w_i^u(\rho_i)) \right\} \partial_{x_i}(u - u^k) d\Omega_\delta^k \right| \\ & \leq C(\alpha + \beta^{1/4}). \end{aligned}$$

The second part in the right-hand side of (16) satisfies

$$\begin{aligned} & \left| \sum_{i=1}^d \int_{\Omega_\delta^k} \left\{ \tilde{\nu}_i^s w_i^s(\rho_i) + \tilde{\nu}_i^u w_i^u(\rho_i) \right\} (u - u^k) d\Omega_\delta^k \right| \\ & \leq C(\alpha + \beta^{1/4}) \|u - u^k\|_{L^2(\Omega_\delta^k)}. \end{aligned}$$

Since the second term in the left-hand side of (16) is nonnegative, we have

$$\|u - u^k\|_{L^2(\Omega_\delta^k)}^2 \leq C(\alpha + \beta^{1/4}),$$

which completes the proof in Case B and finishes the proof of the theorem. □

Theorem 3.3. *Let Ω be a bounded domain with Lipschitz boundary. Suppose $z \in W^{1,1}(\Omega) \cap L^2(\Omega)$. Assume u is the solution of (6) on Ω and \tilde{u} is the solution obtained by (4) and (5). Then we have*

$$(17) \quad \|u - \tilde{u}\|_{L^2(\Omega)}^2 \leq C(\alpha + \beta^{1/4}).$$

Here C is a constant which is dependent on $\|u\|_{W^{1,1}(\Omega)}$ but independent of α and β .

Proof. Since $\tilde{u} = \sum_{k=1}^N E_0^k R_0^k u^k$, we have

$$\|u - \tilde{u}\|_{L^2(\Omega)}^2 = \sum_{k=1}^N \|u - E_0^k R_0^k u^k\|_{L^2(\Omega_0^k)}^2 \leq \sum_{k=1}^N \|u - u^k\|_{L^2(\Omega_0^k)}^2 \leq C(\alpha + \beta^{1/4}),$$

which completes the proof. □

4. Numerical experiments. In this section, some numerical experiments using the proposed algorithm are presented, and we also compare with the results obtained by the classical ROF approach. We focus on the efficiency, scalability and robustness of the algorithm. In the experiments, the noisy images are obtained by adding some Gaussian white noise generated by the Matlab function `imnoise(I, 'gaussian', M, σ^2)`. A standard second-order finite difference method (5-point stencil for 2D and 7-point stencil for 3D) is used to discretize (4) or (6) with a mesh size equals to distance between the two nearest pixels. To measure the quality of the restored images, the peak signal-to-noise ratio (PSNR) is used.

$$(18) \quad \text{PSNR} = 10 \log_{10} \frac{\prod_{i=1}^d N_i}{\sum_j (u_{j,r} - u_j)^2},$$

where $(u_{j,r} - u_j)$ represents the difference of the pixel values between the restored and original images, and N_i is the number of pixels in the i th spatial coordinate direction. Higher values of PSNR means better restoration quality. Typical values of PSNR in lossy image and video compression are between 30dB to 50dB, and acceptable values for wireless transmission quality loss are usually considered to be about 20dB to 25dB.

Our algorithm is implemented on top of the open source package PETSc [3] developed at Argonne National Laboratory. All computations are carried out on the Tianhe 2 supercomputer at the China National Supercomputer Center in Guangzhou. The compute node has a dual six-core Intel Xeon X5650@2.76GHz processor and 24GB of memory.

To compare with the classical ROF model, we implement a parallel solver for (6) using the Newton-Krylov-Schwarz method (NKS) [8] described in Algorithm 2 below, where an inexact Newton method is used as the nonlinear solver and a Krylov subspace method (GMRES) [34] is used to solve the Jacobian system preconditioned by an overlapping domain decomposition method [7]. In Algorithm 2, $F(\mathbf{u}) = \mathbf{0}$ represents the nonlinear system arising from the finite difference discretization of (6) on a rectangular or hexahedral mesh, J_k is the full Jacobian of $F(u)$ at point \mathbf{u}_k , M_k^{-1} is an additive Schwarz preconditioner. The inexactness means that the accuracy of the Jacobian solver is determined by a parameter $\eta \in [0, 1)$ in the sense of

$$(19) \quad \|F(\mathbf{u}_k) + J(\mathbf{u}_k)\mathbf{s}_k\| \leq \eta \|F(\mathbf{u}_k)\|.$$

Algorithm 2 : Newton-Krylov-Schwarz (NKS)

Use the observed image as the initial guess \mathbf{u}_0

For $k = 0, 1, \dots$ until convergence, do

1. Construct the Jacobian matrix J_k
2. Solve the following right-preconditioned Jacobian system inexactly by a Krylov subspace method

$$J_k M_k^{-1} M_k \mathbf{s}_k = -F(\mathbf{u}_k)$$

3. Do a cubic line search to find a step length τ_k
4. Set $\mathbf{u}_{k+1} = \mathbf{u}_k + \tau_k \mathbf{s}_k$

End

The number of subdomains equals to the number of processors and the subdomain solver for the Schwarz method is an incomplete LU factorization (ILU). The relative residual stopping conditions are used for the linear and nonlinear solvers, which are 10^{-4} and 10^{-6} , respectively. The overlap in the Schwarz preconditioner is set to 1. The NKS method is a very powerful parallel solver for nonlinear problems and it has been well studied for lots of problems, for example, problems in fluid dynamics [15, 28] and optimization problems [14, 18].

Note that the subdomain problems (4) in NiOS are also solved with the sequential version of the NKS method on a single processor, where ILU is employed as the subdomain solver. In the rest of the paper, unless otherwise stated, we denote by “Newton” as the total number of Newton iterations for NKS and it also used as the average number of local Newton iterations over all processors for NiOS, “GMRES” as the average number of GMRES iterations per Newton iteration, “Time” as the total compute time in second, and n_p as the number of processors.

4.1. Two-dimensional image denoising. For two-dimensional problems, we test three benchmark gray scale images, boat- 1024×1024 , cameraman- 2048×2048 , and cameraman- 4096×4096 , respectively. The original images have a dynamic range $[0,1]$, and some Gaussian white noises with variance $\sigma^2 = 0.04$ are added to them. For the purpose of comparison, we set the parameters as $\beta = 10^{-4}$ and $\alpha = 0.18$. The computed results are shown in Fig. 3-4. A comparison of the results obtained by the NiOS method and the NKS method is shown in Fig. 4, clearly the restored images of the two methods are very similar. Furthermore, the PSNRs for the two images are almost the same ($\text{PSNR}_{\text{NiOS}} = 28.536962$ and $\text{PSNR}_{\text{NKS}} = 28.536417$). The compute time comparison is in Table 1, which shows that NiOS is much faster than NKS, especially when the number of processors is large, for example, when $n_p = 1024$, for the cameraman- 4096×4096 case, NiOS is 4.5 times faster than NKS.

To further compare the two methods, we plot the difference (or we refer to as the error of the NiOS method) of the two solutions in Fig. 5, and the larger error appears mainly near the inner boundary of the subdomains. Table 2 shows how the error depends on the parameters α and β . We observe that the error decreases with the decrease of α and increases when β decreases. The possible reason that the error increases when β decreases is that Theorem 3.1 gives us an upper bound of $\|u\|_{W^{1,1}(\Omega)}$, but the relationship between $\|u\|_{W^{1,1}(\Omega)}$ and β is still unknown. Thus the constant C in the error estimate (17) may increase as β decreases which leads to the increase of the error. According to the estimate (17), the error is proportional to $\alpha + \beta^{1/4}$ which matches with the numerical results in Table 2. To

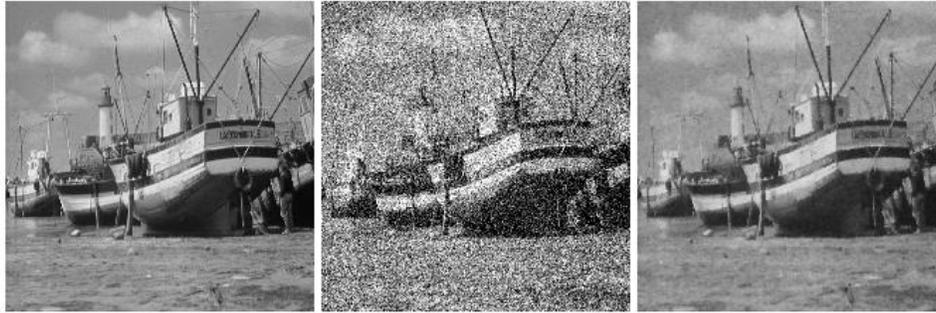


FIGURE 3. The boat-1024 × 1024 image denoising results. From left to right: the original image, the noise image with $\delta^2 = 0.04$, and the restored image by NiOS. Here $\alpha = 0.18$, $\beta = 10^{-4}$, and the PSNR of the restored image is 27.5444.

TABLE 1. The comparison of NiOS and NKS. Here $\beta = 1.0 \times 10^{-4}$, $\alpha = 0.18$, the overlapping size for NiOS is 4. “Sp” refers to the compute time speedup of the NiOS method compared with the NKS method, which is defined as the time of NKS divides the time of NiOS.

n_p	cameraman-2048×2048					cameraman-4096×4096				
	NKS		NiOS		Sp	NKS		NiOS		Sp
	Newton	Time	Newton	Time		Newton	Time	Newton	Time	
32	24	9.8	22	5.0	2.0	30	50.0	26	23.4	2.1
64	27	6.2	21	2.8	2.2	37	35.6	23	12.6	2.8
128	25	3.0	20	1.2	2.5	38	17.9	23	6.5	2.8
256	25	1.7	19	0.6	2.8	33	8.7	21	2.9	3.0
512	24	0.9	19	0.4	2.3	38	5.3	20	1.4	3.8
1024	23	1.2	19	0.3	4.0	38	3.6	20	0.8	4.5

check the convergence rate, we reduce the parameters proportionally, that is if α is reduced by $\frac{1}{2}$, then β should be reduced by $(\frac{1}{2})^4$. Table 3 shows the convergence rate of the algorithm, which matches the estimate from the theory. As mentioned before, β is an artificial parameter used to prevent the singularity at $|\nabla u| = 0$, therefore the value should be chosen as small as possible. As reported in Table 2, the number of nonlinear iterations increases sensibly as β decreases, but the image quality is improved slowly at the same time. So in this paper, we use $\alpha = 0.18$ and $\beta = 1.0 \times 10^{-4}$ unless mentioned otherwise. With these values of α and β , the difference of solutions obtained using NiOS and NKS in relative sense is about 1.0×10^{-3} , which is small enough to indicate that these two images are essentially the same.

The parallel performance of NiOS for the 2D cameraman image case is given in Table 4 and Fig. 6. Table 4 shows that the numbers of Newton and GMRES iterations decrease with the increase of the number of processors (which equals to the number of subdomains). The possible reason is that the subdomain problem becomes smaller when the number of subdomains increases, and this makes the subdomain problem easier to solve. Fig. 6 shows that the parallel performance of the NiOS method is almost ideal (the ideal speedup is defined as: the total compute



FIGURE 4. A comparison of the results obtained by NiOS and NKS. The first row: the clean image (left), the noisy image with $\sigma^2 = 0.04$ (right); second row: the local zoom-in of the figures in the first row; third row: the restored images obtained by NiOS (left) and NKS (right); fourth row: the local zoom-in of the figures in the third row. The PSNR of the restored image with NiOS and NKS are 28.536962 and 28.536417, respectively.

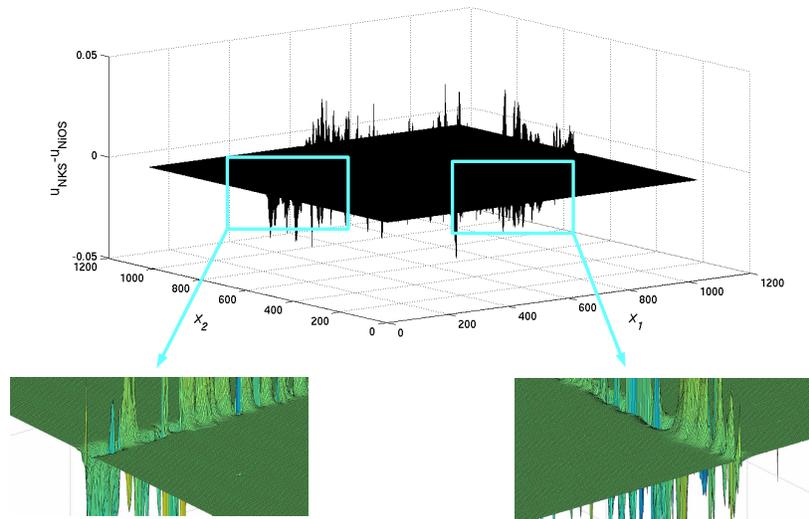


FIGURE 5. The surface plot of the error between NiOS and NKS: $u_{\text{NKS}} - u_{\text{NiOS}}$. Here the image size is 1024×1024 , $\alpha = 0.18$, $\beta = 1.0 \times 10^{-4}$. The overlapping size is 4 and the number of processors is 4.

time is halved as the number of processors is doubled). This is because the algorithm is communication-free and the only factor that influences the parallel efficiency is the unbalance of the subdomain problems, since different subdomain problems have different smoothness properties, so the number of Newton, GMRES iterations, and the total compute time are different as shown in Table 5. The compute time for the fastest subdomain is almost half of that for the slowest subdomain. In all test cases, we consider the time spent by the slowest subdomain as the total compute time. In some of our tests, the total compute time is not exactly halved when the number of processors doubles because the computations on different processors are not always balanced.

4.2. Three-dimensional magnetic resonance image denoising. In medical diagnosis, the three-dimensional magnetic resonance (MR) plays an important role, however, random noise is difficult to avoid in the imaging process and the noise may lead to numerous systematic errors in subsequent applications. For three-dimensional image processing, the computation is more expensive because of the large size. In this section, the performance of the NiOS method is investigated.

The three-dimensional images used for this experiment are from the BrainWeb database¹ [16]. We focus on the restoration of two images: T1-weighted (T1-w) image and T2-weighted (T2-w) image, both of which contain $181 \times 217 \times 181$ voxels. 9% Rician noise is added to the images using the white Gaussian noise as in [17]. Suppose that the pixel values of the images lie in the interval $[0, 1]$. 9% Rician noise means that the Gaussian noise used is equivalent to $\frac{9}{100}\nu$, where ν is the brightest tissue in the image ($\frac{150}{255}$ for T1-w and $\frac{250}{255}$ for T2-w). In the experiments, the PSNR

¹<http://www.bic.mni.mcgill.ca/brainweb/>.

TABLE 2. The error of the NiOS method with respect to the parameters α and β . Here ERROR is the relative error $\|u_{\text{NiOS}} - u_{\text{NKS}}\|_2 / \|u_{\text{true}}\|_2$, u_{NiOS} , u_{NKS} , and u_{true} are the solution of NiOS, NKS, and the clean image, respectively.

β	α	boat-1024×1024				cameraman-2048×2048			
		Newton	GMRES	ERROR	PSNR	Newton	GMRES	ERROR	PSNR
10^{-1}	0.01	2	1	2.1×10^{-7}	15.1	2	1	4.4×10^{-9}	15.4
10^{-1}	0.05	3	1	1.4×10^{-5}	17.4	3	1	2.9×10^{-7}	17.6
10^{-1}	0.1	3	2	6.3×10^{-5}	19.5	3	2	7.1×10^{-6}	19.7
10^{-1}	0.15	4	2	1.3×10^{-4}	21.1	4	2	1.6×10^{-5}	21.2
10^{-1}	0.2	4	2	2.0×10^{-4}	22.3	4	2	3.6×10^{-5}	22.4
10^{-1}	0.25	6	2	2.7×10^{-4}	23.2	12	3	6.1×10^{-5}	23.2
10^{-2}	0.01	2	1	9.3×10^{-7}	15.6	2	1	7.5×10^{-7}	15.6
10^{-2}	0.05	4	2	5.7×10^{-5}	18.8	4	2	3.7×10^{-6}	19.0
10^{-2}	0.1	5	2	2.3×10^{-4}	22.3	5	2	4.5×10^{-5}	22.4
10^{-2}	0.15	6	2	4.3×10^{-4}	24.5	6	2	1.4×10^{-4}	24.6
10^{-2}	0.2	8	3	6.3×10^{-4}	25.7	7	3	2.6×10^{-4}	25.8
10^{-2}	0.25	10	3	8.1×10^{-4}	26.4	8	3	3.8×10^{-4}	26.6
10^{-3}	0.01	3	1	2.2×10^{-6}	15.4	3	1	1.3×10^{-7}	15.7
10^{-3}	0.05	9	2	1.3×10^{-4}	19.3	10	2	1.9×10^{-5}	19.6
10^{-3}	0.1	8	3	4.8×10^{-4}	23.9	8	3	1.8×10^{-4}	24.2
10^{-3}	0.15	10	3	8.7×10^{-4}	26.6	12	3	5.0×10^{-4}	26.9
10^{-3}	0.2	20	4	1.2×10^{-3}	27.4	12	4	8.3×10^{-4}	28.0
10^{-3}	0.25	17	4	1.5×10^{-3}	27.3	14	4	1.1×10^{-3}	28.4
10^{-4}	0.01	9	1	3.2×10^{-6}	15.4	5	2	5.4×10^{-7}	15.7
10^{-4}	0.05	16	3	1.9×10^{-4}	19.4	14	3	5.1×10^{-5}	19.7
10^{-4}	0.1	21	4	7.1×10^{-4}	24.5	20	3	4.1×10^{-4}	24.7
10^{-4}	0.15	19	5	1.3×10^{-3}	27.3	19	5	1.0×10^{-3}	27.9
10^{-4}	0.2	22	7	1.7×10^{-3}	27.5	21	6	1.6×10^{-3}	28.7
10^{-4}	0.25	21	15	2.0×10^{-1}	27.6	24	8	2.1×10^{-3}	28.7

TABLE 3. The error of the NiOS method with respect to the parameters α and β changed proportionally, for example, when α is reduced by $\frac{1}{2}$, β is reduced by $(\frac{1}{2})^4$. “ERROR” has the same definition as in Table 2.

β	α	boat-1024×1024				cameraman-2048×2048			
		Newton	GMRES	ERROR	PSNR	Newton	GMRES	ERROR	PSNR
6.25×10^{-3}	0.1	8	2	2.8×10^{-4}	22.7	6	2	6.4×10^{-5}	22.9
3.9×10^{-4}	0.05	11	2	1.6×10^{-4}	19.3	11	2	3.0×10^{-5}	19.7
2.4×10^{-5}	0.025	14	2	4.1×10^{-5}	16.8	12	2	6.8×10^{-6}	17.1
1.5×10^{-6}	0.0125	17	2	7.6×10^{-6}	15.6	10	2	1.1×10^{-6}	15.9

TABLE 4. Parallel performance of the NiOS algorithm for the 2D cameraman image denoising.

n_p	2048×2048			4096×4096		
	Newton	GMRES	Time(s)	Newton	GMRES	Time(s)
64	21	4	1.7	23	4	7.2
128	20	4	0.8	22	4	3.3
256	20	4	0.4	21	4	1.6
512	19	4	0.2	20	4	0.8
1024	18	4	0.1	20	4	0.4

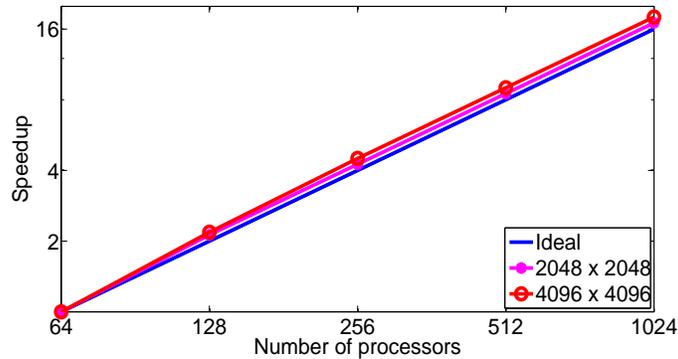


FIGURE 6. The speedup of the NiOS algorithm for the 2D cameraman image denoising problem.

TABLE 5. The detail of the sub-problem solving in NiOS with 24 processors. Image size is 2048×2048. Here “rank” is the processor core ID.

rank	Newton	GMRES	Time	rank	Newton	GMRES	Time
0	11	5	7.3	12	17	5	9.1
1	11	5	7.3	13	14	5	8.4
2	17	5	9.2	14	10	5	6.4
3	19	5	9.8	15	8	5	5.2
4	12	5	7.2	16	14	5	8.0
5	12	5	7.3	17	14	5	8.3
6	12	5	7.3	18	16	5	8.8
7	12	5	7.6	19	16	5	8.8
8	16	5	8.8	20	13	5	7.6
9	17	5	9.0	21	13	5	7.5
10	11	5	6.8	22	16	5	8.5
11	12	5	7.3	23	16	5	6.6

are used as criteria for the quality of image restoration. For the sake of clarity, the PSNR are estimated only in the brain region obtained by removing the background.

Instead of removing the noise slice by slice as in most of the previous works, we use the full three-dimensional model, that is we solve the ROF model (1) in the three-dimensional domain. An example of the noisy image and its partition for parallel computing are shown in Fig. 7. The slices of restored images are shown in Fig. 8 and the three-dimensional reconstructions of the images are shown in Fig. 9. The PSNR values for the noisy T1-w and T2-w images are 26.3706 and 21.9557, and after denoising, the PSNR values are promoted to 30.2275 and 25.7565, respectively.

Table 6 shows a comparison of NiOS and NKS. The results indicate that NiOS saves a lot of compute time compared with NKS, especially when the number of processors is large. Table 7 shows the effect of the overlapping size in the NiOS

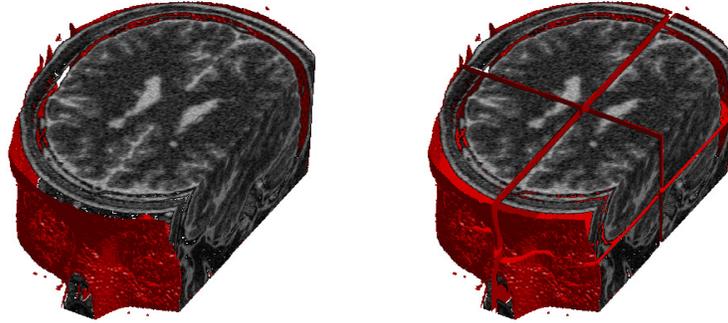


FIGURE 7. A noisy image (left) and an example of the partition of the image into 8 sub-images (right).

TABLE 6. Comparison of the NiOS and NKS for the 3D image denoising. The image size is $181 \times 217 \times 181$. “Sp” refers to the compute time speedup of the NiOS method compared with the NKS method.

n_p	NKS			NiOS			Sp
	Newton	GMRES	Time(s)	Newton	GMRES	Time(s)	
32	35	17	38.5	35	12	22.0	1.8
64	36	17	24.5	31	11	12.4	2.0
128	36	18	12.0	29	10	5.0	2.4
256	36	18	6.6	26	9	3.3	2.0
512	39	18	3.9	24	8	1.4	2.8
1024	35	19	2.4	22	8	0.7	3.4

TABLE 7. The effect of various choices of the overlapping parameter δ in the NiOS method for the 3D image. The number of processors is 64 and the image size is $181 \times 217 \times 181$. The PSNR of the NKS method is 27.84 for this image denoising. Here $\text{DIFF} = \|\mathbf{u}_{\text{NiOS}} - \mathbf{u}_{\text{NKS}}\|_2$ is the difference of the solution of NiOS (\mathbf{u}_{NiOS}) and NKS (\mathbf{u}_{NKS})

δ	Newton	GMRES	Time (s)	PSNR	DIFF
0	14	5	2.7	27.83	5.89
1	14	5	2.8	27.85	1.20
2	14	5	3.1	27.84	0.36
3	14	5	3.4	27.84	0.07

method. We see that a small overlap is good enough in terms of the compute time and image quality. The last column of Table 7 shows that the difference of the solutions of NiOS and NKS decreases with the increase of the overlapping size.

To further understand the newly developed method, we combine the NiOS method and the NKS method by using the solution of the NiOS method as the initial guess of the NKS method. In Table 8, the “First Stage” refers to the NiOS method and the “Second Stage” refers to the NKS method with an initial guess obtained by

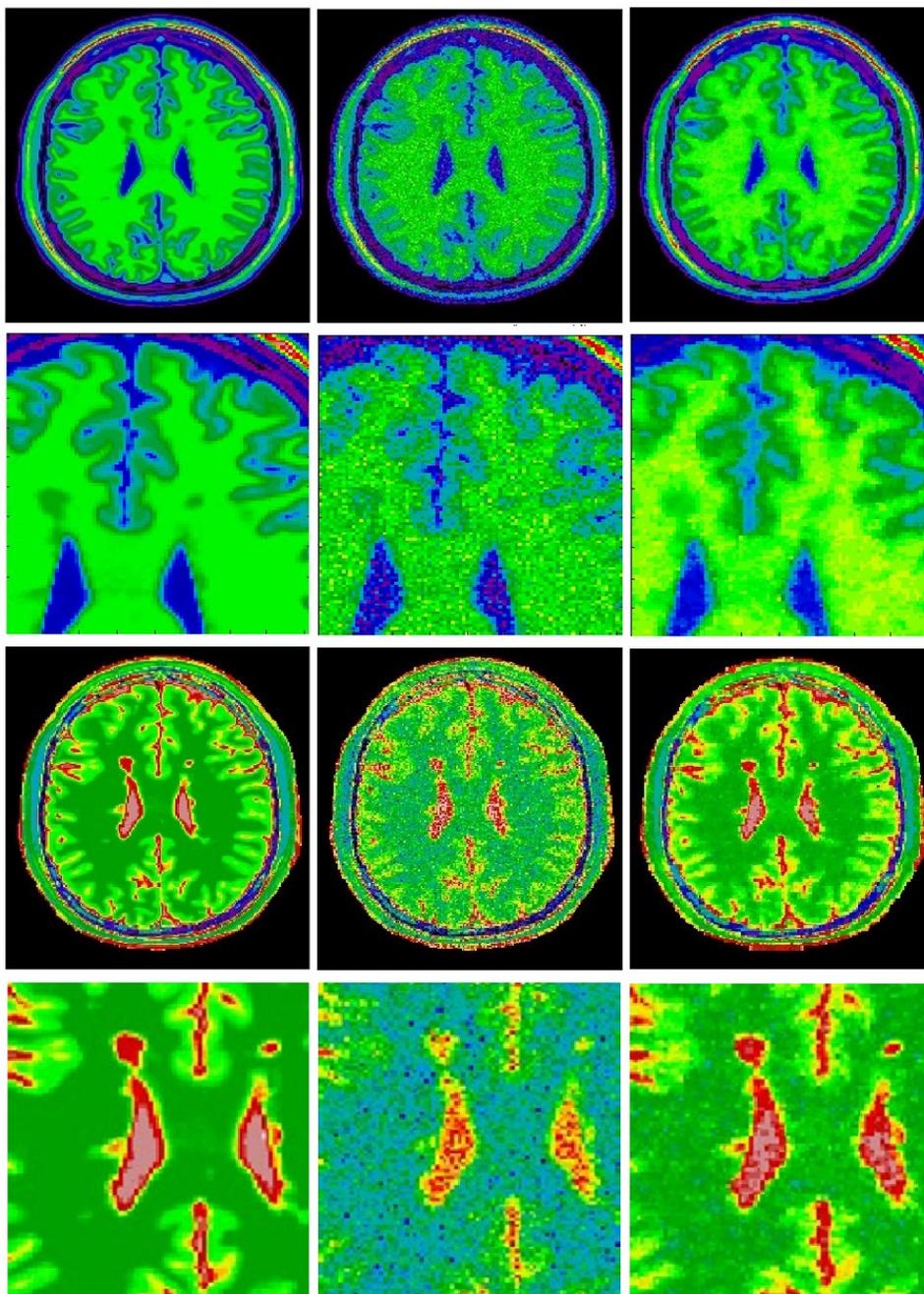


FIGURE 8. Slice 100 of the 3D MR images. From left to right, the first row: the original T1-w image, the T1-w image with a Rician noise at 9% and the restored image; second row: the detailed partial images of the first row images; third row: the original T2-w images, the T2-w image with a Rician noise at 9%, and the restored image. The last row: the detailed partial images of the third row images.

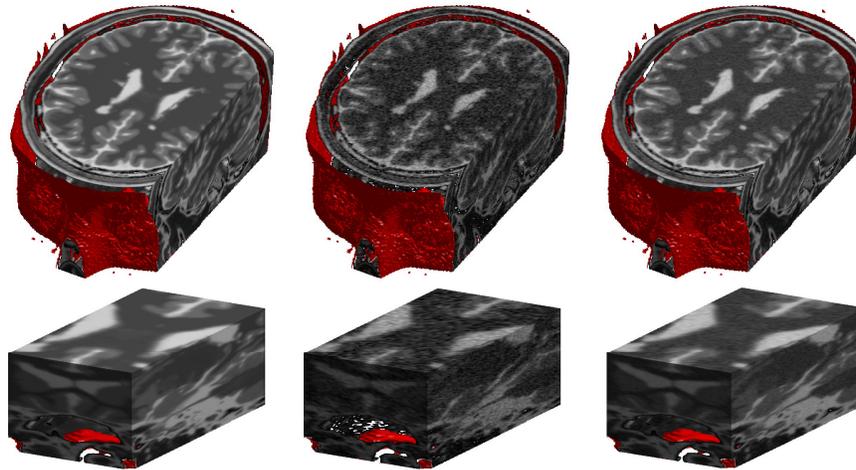


FIGURE 9. The 3D reconstruction of the MR image. From left to right: Top: the clean image, the image with 9% Rician noise, and the restored image; Bottom: the corresponding zoomed images.

TABLE 8. Parallel performance of the NiOS-NKS algorithm for the 3D image denoising. The image size is $217 \times 217 \times 181$.

n_p	First Stage			Second Stage		
	Newton	GMRES	Time	Newton	GMRES	Time
32	15	5	4.48	3	7	2.39
64	14	5	3.5	3	7	1.41
128	15	5	1.71	3	7	0.77
256	14	5	1.11	3	7	0.47
512	14	5	0.45	3	7	0.36
1024	14	5	0.26	3	8	0.42

NiOS. We see that the solution of the NiOS method provides a very good initial guess for the NKS method because the NKS method converges much faster than starting from the noisy image, as compared with the previous experiment.

5. Conclusion. We developed a new parallel non-iterative domain decomposition method for large scale image denoising. Numerical results show that the newly developed method works well for 2D and 3D image denoising problems and a linear speedup is obtained with up to 1024 processors. Traditional domain decomposition methods, whether overlapping or nonoverlapping, require several iterations to find the solution and need at least one coarse grid to achieve linear scalability on parallel computers. The new method requires neither iteration nor coarse grid to obtain the solution and is scalable on parallel computers. A theoretical analysis is provided to show that the error of the method, compared with the classical ROF model, is proportional to $\alpha + \beta^{1/4}$, and the estimate is also confirmed by numerical experiments with two- and three-dimensional images. We have not tested, but would like to mention that the method can easily be combined with other methods, such

as the augmented Lagrangian method, the primal-dual method, the split Bregman iteration, etc.

Acknowledgments. We would like to thank the anonymous referees for their suggestions that helped us improve the paper. The research was supported in part by the National Key R&D Program (No. 2016YFB0200601), the Shenzhen grant under JCYJ20170307165328836, ZDSYS201703031711426 and JCYJ20160331193229720, and the NSFC under 61531166003.

REFERENCES

- [1] A. Asaduzzaman, A. Martinez and A. Sepehri, [A time-efficient image processing algorithm for multicore/manycore parallel computing](#), *Southeastcon, IEEE*, (2015).
- [2] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, Applied Mathematical Sciences, 147. Springer, New York, 2006.
- [3] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. Gropp, B. Smith, D. Karpeyev, D. Kaushik and et al., PETSc Web page, (2019), <https://www.mcs.anl.gov/petsc>.
- [4] M. Basu, [Gaussian-based edge-detection methods-a survey](#), *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **32** (2002), 252–260.
- [5] P. Blomgren, T. F. Chan, P. Mulet, L. Vese and W. L. Wan, Variational PDE models and methods for image processing, *Numerical analysis, Chapman and Hall CRC Research Notes in Mathematics, Boca Raton, FL*, **420** (2000), 43–67.
- [6] A. Buades, B. Coll and J. M. Morel, [Image denoising methods, a new nonlocal principle](#), *SIAM Review*, **52** (2010), 113–147.
- [7] X.-C. Cai, M. Dryja and M. Sarkis, [Restricted additive Schwarz preconditioners with harmonic overlap for symmetric positive definite linear systems](#), *SIAM Journal on Numerical Analysis*, **41** (2003), 1209–1231.
- [8] X.-C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin and D. P. Young, [Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation](#), *SIAM Journal on Scientific Computing*, **19** (1998), 246–265.
- [9] A. Chambolle and P.-L. Lions, [Image recovery via total variation minimization and related problems](#), *Numerische Mathematik*, **76** (1997), 167–188.
- [10] T. F. Chan and J. H. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.
- [11] T. F. Chan, H. M. Zhou and R. H. Chan, *Continuation Method for Total Variation Denoising Problems*, Department of Mathematics, University of California, Los Angeles, 1995.
- [12] H. B. Chang, X.-C. Tai, L.-L. Wang and D. P. Yang, [Convergence rate of overlapping domain decomposition methods for the Rudin-Osher-Fatemi model based on a dual formulation](#), *SIAM Journal on Imaging Sciences*, **8** (2015), 564–591.
- [13] H. B. Chang, X. Q. Zhang, X.-C. Tai and D. P. Yang, [Domain decomposition methods for nonlocal total variation image restoration](#), *Journal of Scientific Computing*, **60** (2014), 79–100.
- [14] R. L. Chen and X.-C. Cai, [Parallel one-shot Lagrange-Newton-Krylov-Schwarz algorithms for shape optimization of steady incompressible flows](#), *SIAM Journal on Scientific Computing*, **34** (2012), B584–B605.
- [15] R. L. Chen, Y. Q. Wu, Z. Z. Yan, Y. B. Zhao and X.-C. Cai, [A parallel domain decomposition method for 3D unsteady incompressible flows at high Reynolds number](#), *Journal of Scientific Computing*, **58** (2014), 275–289.
- [16] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes and A. C. Evans, [Design and construction of a realistic digital brain phantom](#), *IEEE Transactions on Medical Imaging*, **17** (1998), 463–468.
- [17] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann and C. Barillot, [An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images](#), *IEEE Transactions on Medical Imaging*, **27** (2008), 425–441.

- [18] X. M. Deng, X.-C. Cai and J. Zou, [A parallel space-time domain decomposition method for unsteady source inversion problems](#), *Inverse Problems and Imaging*, **9** (2015), 1069–1091.
- [19] D. L. Donoho and I. M. Johnstone, [Adapting to unknown smoothness via wavelet shrinkage](#), *Journal of the American Statistical Association*, **90** (1995), 1200–1224.
- [20] A. Eklund, P. Dufort, D. Forsberg and S. M. LaConte, Medical image processing on the GPU—past, *Present and Future, Medical Image Analysis*, **17** (2013), 1073–1094.
- [21] L. C. Evans, *Partial Differential Equations*, Graduate Studies in Mathematics, 19. American Mathematical Society, Providence, RI, 1998.
- [22] M. Fornasier, A. Langer and C.-B. Schönlieb, [A convergent overlapping domain decomposition method for total variation minimization](#), *Numerische Mathematik*, **116** (2010), 645–685.
- [23] M. Fornasier and C.-B. Schönlieb, [Subspace correction methods for total variation and \$L_1\$ -minimization](#), *SIAM Journal on Numerical Analysis*, **47** (2009), 3397–3428.
- [24] C. Gerhardt, Global regularity of the solutions to the capillarity problem, *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze*, **3** (1976), 157–175.
- [25] M. Hintermüller and A. Langer, [Non-overlapping domain decomposition methods for dual total variation based image denoising](#), *Journal of Scientific Computing*, **62** (2015), 456–481.
- [26] M. Hintermüller and G. Stadler, [An infeasible primal-dual algorithm for total bounded variation-based inf-convolution-type image restoration](#), *SIAM Journal on Scientific Computing*, **28** (2006), 1–23.
- [27] T. Kohlberger, *Variational Domain Decomposition For Parallel Image Processing*, PhD thesis, Universität Mannheim, 2007.
- [28] F. Kong and X.-C. Cai, [A scalable nonlinear fluid-structure interaction solver based on a Schwarz preconditioner with isogeometric unstructured coarse spaces in 3D](#), *Journal of Computational Physics*, **340** (2017), 498–518.
- [29] C.-O. Lee and C. M. Nam, [Primal domain decomposition methods for the total variation minimization, based on dual decomposition](#), *SIAM Journal on Scientific Computing*, **39** (2017), B403–B423.
- [30] F. Malgouyres, [Minimizing the total variation under a general convex constraint for image restoration](#), *IEEE Transactions on Image Processing*, **11** (2002), 1450–1456.
- [31] S. Osher, M. Burger, D. Goldfarb, J. J. Xu and W. T. Yin, [An iterative regularization method for total variation-based image restoration](#), *Multiscale Model. Simul.*, **4** (2005), 460–489.
- [32] G. Pratz and L. Xing, GPU computing in medical physics: A review, *Medical Physics*, **38** (2011), 2685–2697.
- [33] L. I. Rudin, S. Osher and E. Fatemi, [Nonlinear total variation based noise removal algorithms](#), *Physica D: Nonlinear Phenomena*, **60** (1992), 259–268.
- [34] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.
- [35] R. Shams, P. Sadeghi, R. A. Kennedy and R. I. Hartley, [A survey of medical image registration on multicore and the GPU](#), *IEEE Signal Processing Magazine*, **27** (2010), 50–60.
- [36] J. Spruck, [On the existence of a capillary surface with prescribed contact angle](#), *Communications on Pure and Applied Mathematics*, **28** (1975), 189–200.
- [37] N. Ural'tseva, Solvability of the problem of capillaries, *Vestn. Leningr. Univ.*, (1973), 54–64.
- [38] L. A. Vese and T. F. Chan, A multiphase level set framework for image segmentation using the Mumford and Shah model, *International Journal of Computer Vision*, **50** (2002), 271–293.
- [39] C. R. Vogel and M. E. Oman, [Iterative methods for total variation denoising](#), *SIAM Journal on Scientific Computing*, **17** (1996), 227–238.
- [40] J. Xu, X.-C. Tai and L.-L. Wang, [A two-level domain decomposition method for image restoration](#), *Inverse Problem and Imaging*, **4** (2010), 523–545.

Received October 2018; 1st revision May 2019; 2nd revision July 2019.

E-mail address: rl.chen@siat.ac.cn

E-mail address: huangjz@lsec.cc.ac.cn

E-mail address: cai@cs.colorado.edu