Manuscript submitted to
AIMS' Journals
Volume **X**, Number **0X**, XX **200X**

Website: http://AIMsciences.org

pp. **X–XX**

A TWO-LEVEL DOMAIN DECOMPOSITION METHOD FOR IMAGE RESTORATION

Jing Xu

Division of Mathematical Sciences, School of Physical and Mathematical Sciences,
Nanyang Technological University, 637371, Singapore, and
School of Statistics and Mathematics, Zhejiang Gongshang University, 310018, China.

Xue-Cheng Tai

Division of Mathematical Sciences, School of Physical and Mathematical Sciences,
Nanyang Technological University, 637371, Singapore, and
Department of Mathematics, University of Bergen, Bergen, Norway.

Li-Lian Wang

Division of Mathematical Sciences, School of Physical and Mathematical Sciences,
Nanyang Technological University, 637371, Singapore.

(Communicated by the associate editor name)

Abstract. Image restoration has drawn much attention in recent years and a surge of research has been done on variational models and their numerical studies. However, there remains an urgent need to develop fast and robust methods for solving the minimization problems and the underlying nonlinear PDEs to process images of moderate to large size. This paper aims to propose a two-level domain decomposition method, which consists of an overlapping domain decomposition technique and a coarse mesh correction, for directly solving the total variational minimization problems. The iterative algorithm leads to a system of small size and better conditioning in each subspace, and is accelerated with a piecewise linear coarse mesh correction. Various numerical experiments and comparisons demonstrate that the proposed method is fast and robust particularly for images of large size.

1. **Introduction.** Image restoration is one of the fundamental and challenging tasks in image processing [16, 2], and phenomenal advances have been achieved in variational and PDE-based approaches since the seminal work [43]. The ROF model minimizes the total variation (TV) over the space of bounded variation (BV), so it is capable of preserving sharp edges and boundaries with a high quality recovery. More precisely, given a bounded image domain $\Omega \subseteq \mathbb{R}^d\,(d = 1, 2, 3)$, we are interested in the general minimization problem:

$$\min_{u\in BV(\Omega)} \Big\{\alpha \int_\Omega |\nabla u| + \int_\Omega f(u) d\Omega\Big\}, \quad \alpha > 0, \tag{1}$$

2000 *Mathematics Subject Classification.* 68U10, 65M55, 74S20.

Key words and phrases. Overlapping domain decomposition, Coarse mesh correction, Total variation minimization, Image restoration.

This research is supported by Singapore MOE Grant T207B2202, and Singapore NRF2007IDM-IDM002-010. The research of the first author is also partially supported by Youth Foundation of Tianyuan Mathematics, NSF of China with Grant No. 10926037.

where the gradient is in the distributional sense [22], and $f(\cdot)$ is a differentiable functional. The associated Euler-Lagrange equation takes the form

$$-\alpha \mathrm{div}\Big(\frac{\nabla u}{|\nabla u|}\Big) + f'(u) = 0, \tag{2}$$

which is also known as the curvature equation [40]. As the TV model (1)-(2) continues to enjoy applications in diverse areas such as image denoising, debluring and segmentation [43, 12, 37, 54, 5, 63], interface evolution [40, 41], and inverse problems [17], there still exists a great demand for developing fast and robust methods for such minimization problems and nonlinear PDEs, although considerable progress has been made in several directions. Typically, existing methods in the literatures can be classified into the following types:

i. The gradient descent method (cf. [43, 38]): Instead of solving the nonlinear PDE, it involves (2) with an artificial time and minimizes the energy along the gradient descent direct via the evolution of a parabolic equation. This approach is very reliable, but converges considerably slowly.
ii. The lagged diffusivity fixed-point iteration (see, e.g., [1, 55, 56, 57]): It solves the linearized version of the nonlinear steady-state PDE (2) iteratively by treating the nonlinear term $1/|\nabla u|$ explicitly. Various iterative solvers have been considered, but further studies are still needed, in particular, techniques to speed up the outer solvers for large size images.
iii. The dual approach (cf. [10, 5, 15, 14]): It introduces a dual variable (the original unknown function u in (1) is referred to as the primal variable). These methods overcome the non-differentiability of the cost functional in (1). They often lead to more efficient algorithms, and have received increasing interests recently.
iv. Graph-cuts method: It is a well-known technique in image analysis and computer vision [33, 42, 3]. Darbon and Sigelle's work [20] and Chambolle's Markov Random Field based method [11], have introduced the Graph-cuts methods to total variation minimization. Goldfard and Yin [27] have also developed a parametric maximum-flow based method to the original parametric maximum-flow/minimum-cut algorithm in [25] to improve its efficiency. They have also proposed a mixed algorithm for solving the ROF and TV/L1 models more efficiently by combining the Gallo-Grigoriadis-Tarjan algorithm with the divide-and-conquer approach proposed in [20].
v. Additive operator splitting (AOS) scheme: Historically, this type of schemes was first developed for (nonlinear elliptic/parabolic) monotone equation and Navier-Stokes equations in [34, 35]. In image processing applications, the AOS scheme was found to be an efficient way for approximating the Perona-Malik filter [62, 61], especially if symmetry in scale-space is required. The AOS scheme is first order in time, semi-implicit, and unconditionally stable with respect to its time-step [35, 62]. These methods have been applied to a wide range of image processing applications and often lead to very efficient numerical algorithms.
vi. Bregman iteration: Iterative optimization methods based on penalization or Bregman distance [60, 58, 28] have been proposed very recently. In [60, 58], the authors used variable-splitting to separate the L_1 and L_2 terms and then solved an equality constrained optimization problem by penalization and alternative minimization. Bregman iteration for image processing was originated

from [36] and was introduced by Osher et. al. in [39]. It has been extended to wavelet-based denoising [59], nonlinear inverse scale space in [8, 9], and compressed sensing[30, 65]. The basic idea is to transform the equality constrained optimization problem into a series of unconstrained problems using Bregman distance. By combining the variable-splitting and Bregman iteration, Goldstein et. al. obtained split Bregman method in [28] which is particularly efficient for L_1 regularized problems, e.g., TV restoration.

vii. Augmented Lagrangian method [26]: It was proposed in [51] for total variation image restoration. It has many advantages over other methods such as penalty method [4]. As only linear problems need to be solved during the iterations, FFT can be applied to get extremely efficient implementations. In addition, the augmented Lagrangian approach provides close connections to dual methods and split Bregman iteration [65, 51].

viii. Multigrid method [52, 53, 64]: It is one of the most powerful numerical methods for solving some linear and nonlinear partial differential equations. In [18, 31], the linear algebraic multigrid method [46] was adopted for solving the above PDE in each (outer) step of a fixed iteration, while [44] attempted to use the standard multigrid methods with a non-standard and somewhat global smoother. Recently, nonlinear multigrid methods based on the subspace correction approach of [52] have been introduced to image processing in [19, 13]. Numerical experiments indicate their overwhelming numerical potentials.

These methods have been widely used for image processing, and their strength and weakness have also been observed from real applications. Dual methods and Bregman iterations are fast, but they are under intensive investigation for the applications to more general image processing problems. Graph-cut approaches are usually fast, but they can be only applied to a special class of problems and could also have matriculation errors. The AOS and multigrid methods also have limitations in the models that they can be applied.

The purpose of this paper is to propose a fast solver based on overlapping domain decomposition and a coarse mesh correction for image processing tasks. Our aim is to demonstrate several essential advantages of the proposed method. More precisely,

i. This method can be used for very general variational-based image processing problems. Indeed, based on this notion, one can easily apply the existing solvers to the minimization problem of a given task by solving a sequence of subdomain problems of smaller scale.

ii. In practice, the original problem, e.g., large size 3D data processing, could be too large, which induces difficulties in applying a given solver. By splitting a large problem into many smaller sub-problems, we could apply the given solver.

iii. The proposed method can save CPU time cost. The gain is significant with an implementation of efficient and relatively accurate subdomain solvers,

iv. The proposed method is well-suited for distributed-memory parallel computers, hence it can be sped up by parallel implementation.

It is known that domain decomposition (DD) methods are powerful iterative methods for solving partial differential equations [7, 21, 29, 45, 64]. Some recent progress has shown that DD methods are also efficient for some nonlinear elliptic problems and some convex minimization problems [49, 48, 50, 52] with mesh independent convergence. So far, it is still unknown that one can use domain decomposition methods for the ROF model. Some recent efforts have been devoted

to study these problems [47, 32, 24, 23]. For simplicity of presentation, we propose and test this method on the ROF model, see (1), and present the details of the implementation. We provide ample numerical results to show its capability in processing images of large size with saving in CPU time and memory. Once again, the essence of the method is to regard domain decomposition method as a space decomposition technique. The original minimization problem related to ROF is reduced to some sub-minimization problems with smaller size over the sub-domains. If the sub-minimization problems are solved exactly, the convergence of the generated sequence is trivial to prove. Due to the degeneracy of the nonlinear equation of ROF, it is not convincing that we will be able to prove the convergence rate for the numerical solutions.

The rest of the paper is organized as follows. In section 2, we present the domain decomposition algorithm under a general framework of the subspace correction method. We describe the two-level method in section 3, and we give the detailed implementation for the ROF model in section 4. We provide various numerical results to demonstrate the merits of the proposed methods in section 5. We conclude the paper some discussions.

2. Domain decomposition based subspace correction method.

We put the method in a more general setting and start with the description of the subspace correction algorithm of [52].

Given a reflexive Banach space V and a convex, Gateaux differentiable functional $F : V \to \mathbb{R}$, we consider the minimization problem:

$$\min_{u\in V} F(u). \tag{3}$$

Under the notion of space correction, we first decompose the space V into a sum of smaller subspaces:

$$V = V_1 + V_2 + \cdots + V_m, \tag{4}$$

which means that for any $v \in V$, there exists $v_j \in V_j$ such that $v = \sum_{i=1}^{m} v_j$.

Following the framework of [64] for linear problems, we solve a finite sequence of sub-minimization problems over the subspaces:

$$\min_{e\in V_j} F(u^n + e), \tag{5}$$

where u^n denotes a previous approximation, to resolve (3). Two types of subspace correction methods based on (4)-(5), known as the parallel subspace correction (PSC) and successive subspace correction (SSC) method, were proposed in [64, 52]. Here, we adopt the latter, which can be described as follows:

Algorithm SSC. Choose an initial value $u^0 \in V$.

For $n = 0$,

while $j = 1, \cdots, m$ **do**

Find $e_j^n \in V_j$ such that

$$F(u^{n+(j-1)/m} + e_j^n) \le F(u^{n+(j-1)/m} + v_j), \quad \forall v_j \in V_j.$$

Set

$$u^{n+j/m} = u^{n+(j-1)/m} + e_j^n.$$

end

Go to next iteration for n.

In the following, we apply the algorithm to the (regularized) ROF denoising model with the cost functional:

$$F(u) = \alpha \int_\Omega \sqrt{u_x^2 + u_y^2 + \beta}\, dxdy + \frac{1}{2}\int_\Omega |u - z|^2\, dxdy, \quad \alpha, \beta > 0, \tag{6}$$

where z is a given noisy image defined on $\Omega = (0,1) \times (0,1)$. Here, the TV-term is regularized so that F is differentiable and it also avoids the division by zero in the corresponding Euler-Lagrange equation:

$$u - \alpha \mathrm{div}\left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}\right) = z, \tag{7}$$

with a homogenous Neumann boundary condition $\partial u/\partial \mathbf{n} = 0$ along the boundary. Recall that the lagged diffusivity fixed-point iteration (cf. [56]) for (7) is to solve the linearized equation

$$u^{k+1} - \alpha \mathrm{div}\left(\frac{\nabla u^{k+1}}{\sqrt{|\nabla u^k|^2 + \beta}}\right) = z, \quad k = 0, 1, \cdots, \tag{8}$$

with the initial value u^0. We see that each iteration involves all the pixel values in the image domain, so it will be costly and usually the system is not in good conditioning when the size of images is large. The domain decomposition based SSC algorithm will overcome the difficulties by breaking down the whole problem into sub-problems of much smaller size.

In the first place, we use an overlapping domain decomposition to decompose the solution space $V = BV(\Omega)$. More precisely, we partition Ω into m overlapping subdomains

$$\Omega = \bigcup_{j=1}^{m} \Omega_j, \quad \Omega_j \cap \Omega_k \neq \emptyset, \quad k \neq j. \tag{9}$$

For clarity, the subdomain Ω_j is colored with a color j, and Ω_j consists of m_j subdomains (assumed to be "blocks" for simplicity), which are not intersected. Hence, the total number of blocks that cover Ω is

$$M := \sum_{j=1}^{m} m_j. \tag{10}$$

Figure 1 illustrates schematically the decomposition of Ω into four colored subdomains with 25 blocks.

Based on the above domain decomposition, we decompose the space $V = BV(\Omega)$ as

$$V = \sum_{j=1}^{m} V_j, \quad V_j = BV_0(\Omega_j), \tag{11}$$

where $BV_0(\Omega_j)$ denotes the subspace of $BV(\Omega_j)$ with zero traces on the "interior" boundaries $\partial\Omega_j \backslash \partial\Omega$. Applying the SSC algorithm to the TV-denoising model leads to an iterative method.

Given an initial value $u^0 \in V$, Algorithm SSC needs us to solve u^n from

$$\begin{cases} F\left(u^{n+\frac{j-1}{m}} + e_j^n\right) \leq F\left(u^{n+\frac{j-1}{m}} + v_j\right), \quad \forall v_j \in V_j = BV_0(\Omega_j), \\ u^{n+\frac{j}{m}} = u^{n+\frac{j-1}{m}} + e_j^n, \quad 1 \leq j \leq m. \end{cases} \tag{12}$$

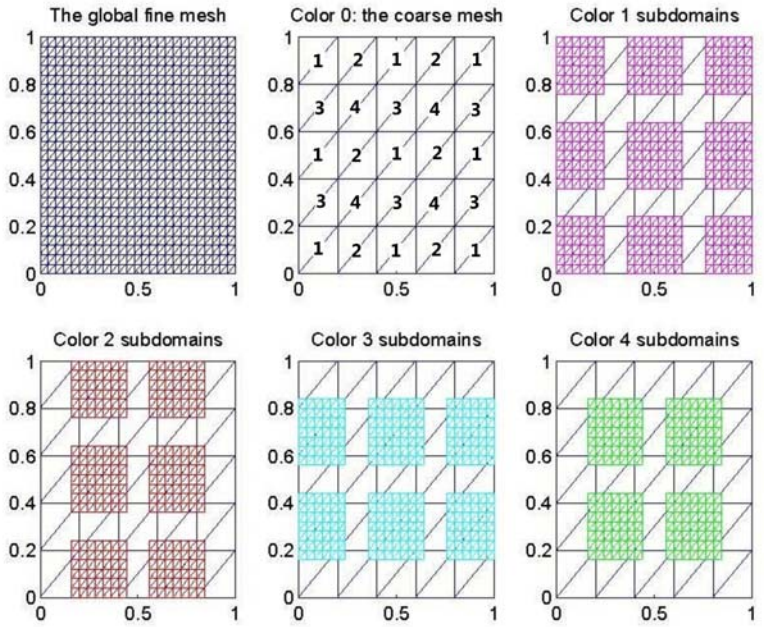


FIGURE 1. Schematic illustration of the coloring of the subdomains, and fine/coarse meshes on $\Omega = (0,1)^2$. This corresponds to the decomposition: $V^h = V_0^H + \sum_{i=1}^4 V_i^h$ with $H = 5h$ and in (10), $m = 4$, $m_1 = 9, m_2 = 6, m_3 = 6, m_4 = 4$, and $M = 25$.

Here, we notice that e_j^n is the solution of the subproblem over Ω_j. It is also easy to see that $u^{n+\frac{j}{m}}$ satisfies the associated Euler-Lagrange equations for $1 \le j \le m$,

$$\begin{cases} -\alpha \mathrm{div}\Big(\frac{\nabla u^{n+\frac{j}{m}}}{\sqrt{|\nabla u^{n+\frac{j}{m}}|^2+\beta}}\Big) + u^{n+\frac{j}{m}} = z, & \text{in } \Omega_j, \\ \frac{\partial u^{n+\frac{j}{m}}}{\partial n} = 0, & \text{on } \partial\Omega_j \cap \partial\Omega, \\ u^{n+\frac{j}{m}} = u^{n+\frac{j-1}{m}}, & \text{on } \partial\Omega_j \backslash \partial\Omega. \end{cases} \tag{13}$$

Outside Ω_j, we have $u^{n+\frac{j}{m}} = u^{n+\frac{j-1}{m}}$. Thus, there is no need to solve $u^{n+\frac{j-1}{m}}$ outside Ω_j. As the subdomain Ω_j may contain many disjoint "block", the values of $u^{n+\frac{j}{m}}$ can be obtained in parallel in these "blocks" by solving (13).

More details on the disretization of (13) will be described in the forthcoming section.

3. The two-level domain decomposition method.

In this section, we build another ingredient, i.e., a coarse mesh correction, into the previous domain decomposition method. For clarity of presentation, we introduce the coarse mesh solver in the finite element setting. Similar explanations are also valid for the finite difference approximations.

We first partition the domain Ω into a coarse mesh $\{\mathcal{T}_H\}$ with a mesh size H, and then refine it into a fine mesh partition $\{\mathcal{T}_h\}$ with a mesh size $h < H$. Assume that both the coarse mesh and the fine mesh are shape-regular, and let $\{D_i\}_{i=1}^m$ be

a non-overlapping domain decomposition for Ω and each D_i is the union of some coarse mesh elements (see Figure 1).

Let $V := V(\Omega)$ be the space to be specified later. Let $S^H \subset V(\Omega)$ and $S^h \subset V(\Omega)$ be the continuous, piecewise linear finite element spaces, over the H-level and h-level subdivisions of Ω respectively. More specifically,

$$\begin{aligned} S^H &= \big\{v \in C^1(\Omega) \ : \ v|_{\mathcal{T}} \in P_1(\mathcal{T}), \ \ \forall \mathcal{T} \in \mathcal{T}_H\big\}, \\ S^h &= \big\{v \in C^1(\Omega) \ : \ v|_{\mathcal{T}} \in P_1(\mathcal{T}), \ \ \forall \mathcal{T} \in \mathcal{T}_h\big\}. \end{aligned} \tag{14}$$

For each D_i, we consider an enlarged sub-domain $\Omega_i = D_i^\delta$ consisting of elements $\mathcal{T} \in \mathcal{T}_h$ with $\text{dist}(\mathcal{T}, D_i) \le \delta$. The union of Ω_i covers $\overline{\Omega}$ with overlaps of size δ. Let us denote the piecewise linear finite element space with zero traces on the boundaries $\partial\Omega_i \backslash \partial\Omega$ as $S^h(\Omega_i)$. Then one can show that

$$S^h = \sum S^h(\Omega_i) \quad \text{and} \quad S^h = S^H + \sum S^h(\Omega_i). \tag{15}$$

For the overlapping subdomains, assume that there exist m colors such that each subdomain Ω_i can be marked with one color, and the subdomains with the same color will not intersect with each other. For suitable overlaps, one can always choose $m = 2$ if $d = 1$; $m = 4$ if $d \ge 2$; $m \ge 8$ if $d = 3$. Let Ω_i be the union of the subdomains with the i^{th} color, and define

$$V_i^h = \big\{v \in S^h \ : \ v(x) = 0, \ x \notin \Omega_i\big\}, \ i = 1, 2, \cdots, m.$$

By denoting subspaces $V_0^H = S^H$, $V^h = S^h$, we get from (15) that

$$a). \quad V^h = \sum_{i=1}^m V_i^h \quad \text{and} \quad b). \quad V^h = V_0^H + \sum_{i=1}^m V_i^h. \tag{16}$$

Note that the summation index is from 0 to m instead of from 1 to m when the coarse mesh is added.

For a better understanding of the above slightly abstract setting, we present below two illustrative examples.

Example I: overlapping

Divide Ω into only four nonoverlapping subdomains (see Figure 2 (left)):

$$D_1 = (0,1/4)\times(0,1/4)\cup(1/2,3/4)\times(0,1/4)\cup(0,1/4)\times(1/2,3/4)\cup(1/2,3/4)\times(1/2,3/4)$$

and get the overlapping subdomains corresponding

$$\Omega_1 = (0,5/16)\times(0,5/16)\cup(7/16,13/16)\times(0,5/16)\cup(0,5/16)\times(7/16,13/16)\cup(7/16,13/16)\times(7/16,13/16)$$

where $\delta = 1/16$.

Example II: A coarse mesh correction to space decomposition. We consider the simple unit square domain $\Omega = (0,1) \times (0,1)$ and a uniform triangulation $\mathcal{T}_H(\Omega) = \{\tau\}$ and piecewise linear finite element spaces. If we take $H = 4h$, then the values of the 2D basis function may be denoted by matrix I_H^h (called the correction

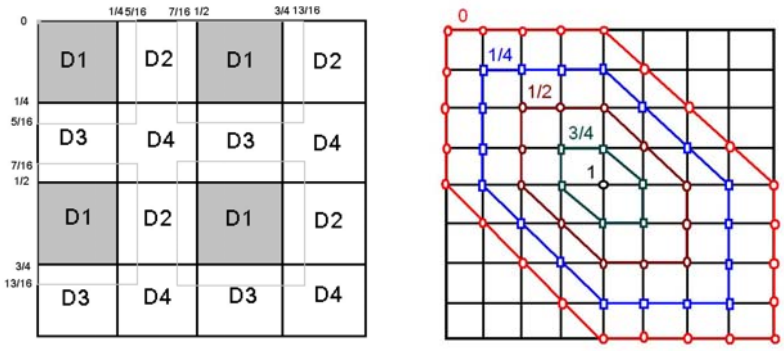


FIGURE 2. Left: an overlapping decomposition. Right: The finite element basis at the center node. The weights from center towards boundary are 1 (marked by black '∘'), 3/4 (marked by green '□' along the innermost hexagon), 1/2 (marked by brown '∘'), 1/4 (marked by blue '□'), 0 (marked by red '∘' along the innermost hexagon)

operator), which takes the values (see Figure 2 (right)):

$$I_H^h = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{17}$$

If we apply Algorithm SSC to decomposition (16) with the coarse mesh, we will get the following domain decomposition algorithm:

Algorithm DDC. Choose an initial value $u_h^0 \in V^h$.

For $n = 0$,

Set $\tilde{u}_h^n = u_h^n$,

while $j = 1, \cdots, m$ **do**

Find $e_{h,j}^n \in V_j^h$ such that

$$e_{h,j}^n = \arg \min_{v_h \in V_j^h} F(\tilde{u}_h^{n+(j-1)/m} + v_h).$$

Set

$$\tilde{u}_h^{n+j/m} = \tilde{u}_h^{n+(j-1)/m} + e_{h,j}^n.$$

end

Find $e_{H,0}^n \in V_0^H$ such that

$$e_{H,0}^n = \arg \min_{v_H \in V_0^H} F(\tilde{u}_h^{n+1} + v_H).$$

Set

$$u_h^{n+1} = \tilde{u}_h^{n+1} + I_H^h e_{H,0}^n.$$

Go to next iteration for n.

It is seen that the above iteration algorithm requires to solve a sequence of the minimization problems over the subspaces/subdomains. For the TV-denoising problem (6), the prototypical variational formulation of the sub-minimization problem is

$$\begin{cases} \text{Given } u^n, \text{ find } e^n \in \mathcal{V} \text{ such that} \\ \alpha\left(\dfrac{\nabla(u^n + e^n)}{\sqrt{|\nabla(u^n + e^n)|^2 + \beta}}, \nabla v\right) + (u^n + e^n - z, v) = 0, \quad \forall v \in \mathcal{V}, \end{cases} \tag{18}$$

where $(\cdot, \cdot)$ is the $L^2(\Omega)$ inner product, and $\mathcal{V}$ is the finite dimensional space V_j^h or V_0^H.

In real implementations, we linearize (18). For $\mathcal{V} = V_j^h$, since $\tilde{u}_h^{n+j/m} = \tilde{u}_h^{n+(j-1)/m} + e_{h,j}^n$, we solve the problem:

$$\alpha\left(\frac{\nabla \tilde{u}_h^{n+j/m}}{\sqrt{|\nabla \tilde{u}_h^{n+(j-1)/m}|^2 + \beta}}, \nabla v\right) + (\tilde{u}_h^{n+j/m} - z, v) = 0, \quad \forall v \in V_j^h, \tag{19}$$

to obtain $\tilde{u}_h^{n+j/m}$ for all $1 \le j \le m$. For $\mathcal{V} = V_0^H$, the linearized problem for $e_{H,0}^n$ is

$$\alpha\left(\frac{\nabla(\tilde{u}_h^{n+1} + e_{H,0}^n)}{\sqrt{|\nabla \tilde{u}_h^{n+1}|^2 + \beta}}, \nabla v\right) + \left(\tilde{u}_h^{n+1} + e_{H,0}^n - z, v\right) = 0, \quad \forall v \in V_0^H. \tag{20}$$

However, a little care has to be taken for the transition between the coarse mesh and fine mesh, and the details will be presented in the forthcoming section.

4. Numerical discrete algorithm for TV denoising.

We next present the full two-level algorithm formulated in the previous section for the TV-denoising model. We partition the image domain $\Omega = (0, 1)^2$ into $N \times N$ uniform cells with mesh size $h = 1/N$. The cell centers are

$$(x_i, y_j) = \big((i-1)h, (j-1)h\big), \quad 1 \le i, j \le N + 1. \tag{21}$$

Hereafter, let $z_{i,j}$ be the pixel value of the original image z at (x_i, y_j). It is known that using proper quadrature rules, finite element methods reduce to finite difference approximations on uniform grids. Hereafter, we shall adopt finite difference to discretize (19) and (20). Let $u_{i,j}$ be the finite difference solution at (x_i, y_j). Denote

$$\delta_x^{\pm} u_{i,j} = \pm(u_{i\pm1,j} - u_{i,j}), \quad \delta_y^{\pm} u_{i,j} = \pm(u_{i,j\pm1} - u_{i,j}).$$

$$\delta_x^{c} u_{i,j} = u_{i+1,j} - u_{i-1,j}, \quad \delta_y^{c} u_{i,j} = u_{i,j+1} - u_{i,j-1}.$$

The finite difference approximation of (19) is

$$\begin{aligned} u_{l,k} - \alpha_h \Bigg\{ \delta_x^- \left[\frac{\delta_x^+ u_{l,k}}{\sqrt{(\delta_x^+ u_{l,k})^2 + (\delta_y^c u_{l,k})^2 + \beta_h}} \right] \\ + \delta_y^- \left[\frac{\delta_y^+ u_{l,k}}{\sqrt{(\delta_x^c u_{l,k})^2 + (\delta_y^+ u_{l,k})^2 + \beta_h}} \right] \Bigg\} = z_{l,k}, \end{aligned} \tag{22}$$

where $\alpha_h = \alpha/h$ and $\beta_h = h^2\beta$. The one-sided second-order finite differences are used to treat the Neumann boundary conditions, say at $x = 0$:

$$u_{0,k} = \frac{4}{3} u_{1,k} - \frac{1}{3} u_{2,k}. \tag{23}$$

Boundary conditions are also needed when evaluating δ_x^c and δ_y^c at the boundary nodes.

We now turn to the coarse mesh problem (20). Firstly we note that (20) can be written in the form:

$$\alpha \left(a^n \nabla e_{H,0}^n, \nabla v\right) + (e_{H,0}^n, v) = \left(z - \tilde{u}_h^{n+1}, v\right) - \alpha \left(a^n \nabla \tilde{u}_h^{n+1}, \nabla v\right), \quad \forall v \in V_0^H. \tag{24}$$

with

$$a^n = (|\nabla \tilde{u}_h^{n+1}|^2 + \beta)^{-1/2}.$$

As before, we need to define a restriction operator for the explanation of the algorithm. For any given $r \in S^h$, we define $I_h^H r \in S^H$ such that

$$(I_h^H r, v) = (r, v), \quad \forall v \in S^H.$$

Given a $r \in S^h$, $I_h^H r$ can be obtained numerically by a proper summation of r multiplied with the matrix I_H^h defined in (17) over the support of the finite element basis functions. Analysis in [49] showed that we could solve the subproblems approximately. So we shall use the following finite difference scheme to solve (20)

approximately:

$$\begin{aligned} e_{L,K} - \alpha_H \Bigg\{ \delta_x^- \left[\frac{\delta_x^+ e_{L,K}}{\sqrt{(\delta_x^+ u_{L,K})^2 + (\delta_y^c u_{L,K})^2 + \beta_H}} \right] \\ + \delta_y^- \left[\frac{\delta_y^+ e_{L,K}}{\sqrt{(\delta_x^c u_{L,K})^2 + (\delta_y^+ u_{L,K})^2 + \beta_H}} \right] \Bigg\} = \\ I_h^H \Bigg[z_{l,k} - u_{l,k} + \alpha_h \Bigg\{ \delta_x^- \left[\frac{\delta_x^+ u_{l,k}}{\sqrt{(\delta_x^+ u_{l,k})^2 + (\delta_y^c u_{l,k})^2 + \beta_h}} \right] \\ + \delta_y^- \left[\frac{\delta_y^+ u_{l,k}}{\sqrt{(\delta_x^c u_{l,k})^2 + (\delta_y^+ u_{l,k})^2 + \beta_h}} \right] \Bigg\} \Bigg], \end{aligned} \tag{25}$$

where $\alpha_H = \alpha/H$ and $\beta_H = H^2\beta$. Formly, the above system can be written as

$$\tilde{A}_H e_H = I_h^H \big(z - A_h u_h\big), \tag{26}$$

where $\{\tilde{A}_H, A_h\}$ are the coefficient matrices of the systems (22) and (25), respectively, and $\{e_H, z, u_h\}$ are vectors of the pixel values. The matrix $\tilde{A}_H$ is an approximation of the system matrix for finite element equation (24), that is, we have replaced a^n by the known data a^{n-1}, and also used the values at the coarse grid points.

We summarize the above algorithm as follows.

Algorithm DDC-TV. Choose an initial value $u_h^0 \in V^h$.

For $n = 0$,

Set $\tilde{u}_h^n = u_h^n$,

while $j = 1, \cdots, m$ **do**

  Solve (22):

$$A_{h,j} \tilde{u}_h^{n+j/m} = z_j.$$

end

Solve (25):

$$\tilde{A}_H e_{H,0}^n = I_h^H (z - A_h \tilde{u}_h^{n+1}).$$

Set

$$u_h^{n+1} = \tilde{u}_h^{n+1} + I_H^h e_{H,0}^n.$$

Go to next iteration for n.

5. **Numerical results.** We present in this section various numerical results to demonstrate the efficiency of the proposed domain decomposition algorithms without or with a coarse domain correction, denoted by DD and DDC in short, respectively. Their performance is assessed by comparing with the naive lagged diffusivity fixed-point iteration (i.e., (8), denoted by TV) in terms of convergence, recovery of peak signal-to-noise ratio (PSNR) and computational time. We remark that to the best of our knowledge, these algorithms have not been applied to the image restoration problems before. Hence, it is interesting to see some good results and particularly how the methods can be applied to restore images of large size.

Hereafter, assume that the pixel values of all images lie in the interval $[0, 255]$, and the Gaussian white noise is added by the normal imnoise function `imnoise(I,'gaussian', M, σ)` (i.e., the mean M and variance σ) in `Matlab`. In our tests, we use PSNR [6] as a criteria for the quality of restoration. This quantity is usually expressed in terms of the logarithmic decibel scale:

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{n_1 n_2} \sum_{i,j} (u_{i,j} - z_{i,j})^2}, \tag{27}$$

where $\{u_{i,j} - z_{i,j}\}$ are the differences of the pixel values between the restored and original images of size $n_1 \times n_2$. Typical values for the PSNR in lossy image and video compression are between 30dB and 50dB (the higher implies the better). Acceptable values for wireless transmission quality loss are considered to be about 20dB to 25dB. We shall also use the relative dynamic error between two consecutive iterations:

$$\frac{\|u^k - u^{k-1}\|_2}{\|u^k\|_2} < \epsilon, \tag{28}$$

for a prescribed tolerance ϵ, as the stopping rule.

We test the methods on three typical images: lena-512×512, boat-1024×1024, and cow-2048×2048. All the computations are done in `Matlab` on an IBM server with 2.93 GHz, 8 Intel(R) Xeon(R) Quad-Core CPU and 128GB RAM. In the first two sets of experiments, we fix $\beta = \varepsilon = 10^{-4}$, and choose the mean value $M = 0$ and the variance $\sigma = 0.04$ for the noise level, whose signal-to-noise ratio (SNR) is roughly between 8.8 to 9.1. We compare the methods with different α, subdomain size d (pixels) and overlapping size δ (pixels), and show their performance with respect to the iteration number k and computational time T. Finally, we test the methods with smaller regularization constant β, and other noise levels.

We shall see that the proposed methods lead to significant time and memory saving. Moreover, they are not sensitive to the image size, and the choice of the intrinsic parameters d and δ can be fairly relaxed. Hence, they provide fast and robust means to process images in particular of large size.

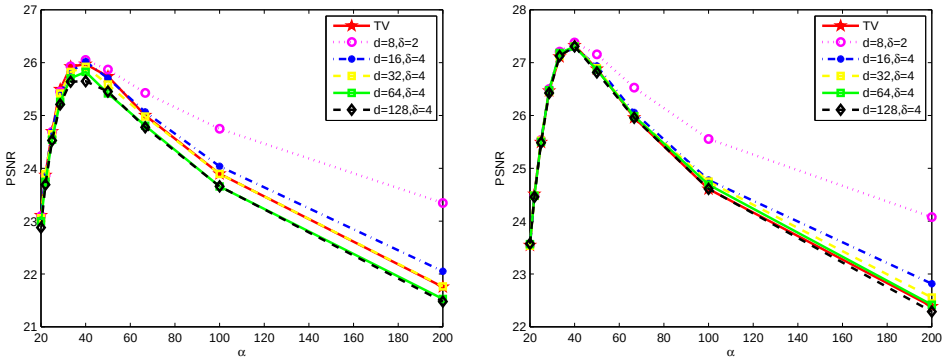


FIGURE 3. The impact of α to PSNR: lena-512×512 (left) and boat-1024×1024 (right).

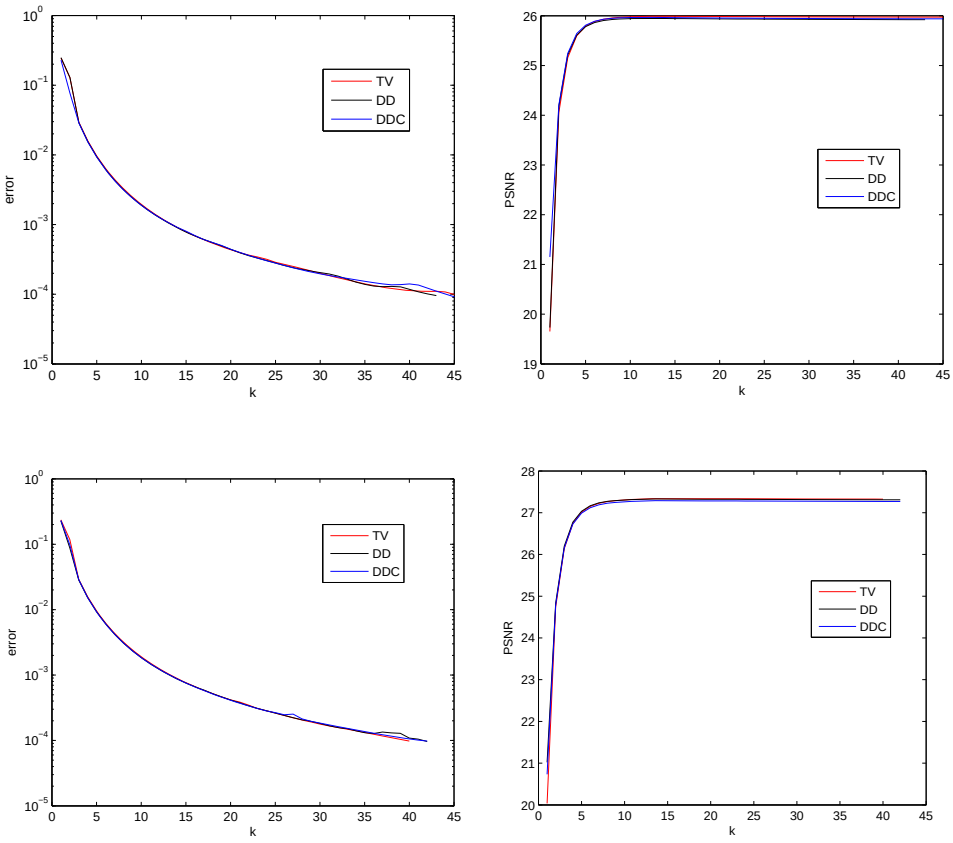


FIGURE 4. The dynamic error (cf. (28)) (left column) and the PSNR (right column) at iteration step k for three methods (for DD and DDC, $d = 32$ and $\delta = 4$) with lena-512×512 (upper row) and boat-1024×1024 (lower row).

5.1. **Convergence.** To set up a more quantified and reasonable rule for comparison, we fix $\beta = \varepsilon = 10^{-4}$, and first identify a suitable α in the TV model with larger PSNR values (i.e, a better restoration). We plot in Figure 3 the PSNR against α obtained by TV (i.e., (8)) and the DD (i.e., (13)) with different subdomain and overlapping sizes for lena-512×512 and boat-1024×1024. For both cases, a good choice is α between 30 and 50. Hereafter, we fix $\alpha = 40$. We also observe from Figure 4 that the PSNR reaches the "maximum" values after about ten iterations for TV, DD and DDC.

In Figure 4, we plot the dynamic error history of the three methods and PSNR values against the iteration steps. We see that the DD and DDC exhibit a convergence behavior similar to that of the fixed-point iteration. Hence, the domain decomposition method produces as good quality as the classical TV restoration via local operations and/or some global corrections. In Figure 5, we plot the computed solution for the first iteration for the boat-1024×1024 image. We have used $d = 32$ and $\delta = 4$. These plots visualize the local step-by-step denoising effect and the recovery through overlapping subdomains.

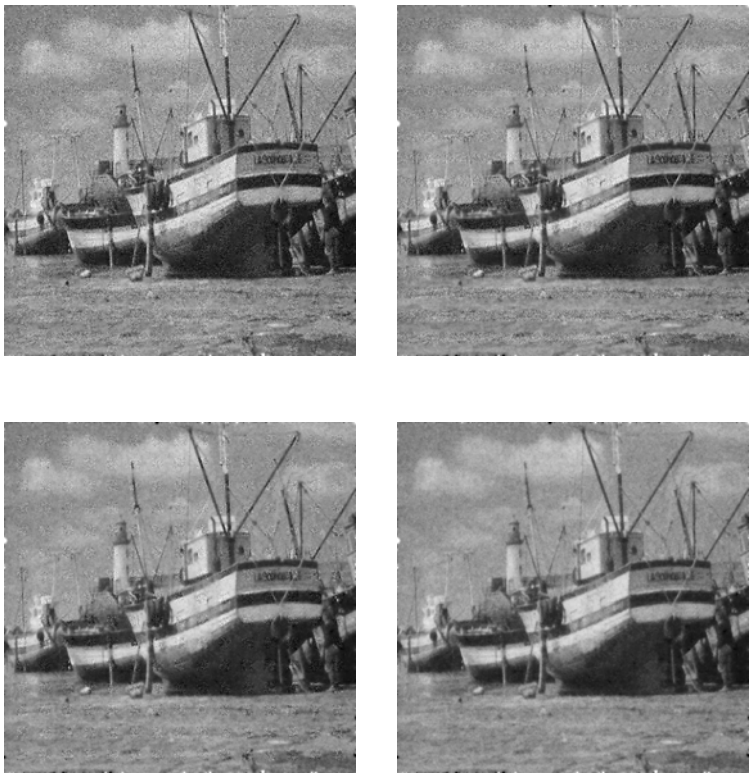


FIGURE 5. The intermediate results of DD for the boat image for $u^{n+\frac{j}{m}}$ with $n = 0, j = 1, 2, 3, 4$. The subdomains are painted with 4 colors. From left to right: the intermediate image obtained by solving (13) for $n = 0, j = 1, 2, 3, 4$. Here, the image size: 1024×1024, the subdomain size: 32, and the overlapping size: 4.

TABLE 1. Different overlapping size with parameters $\epsilon = 10^{-4}$, $\sigma = 0.04$, $\alpha = 40$ and $\beta = 10^{-4}$.

image	d	δ	k	PSNR	Time	image	d	δ	k	PSNR	Time
lena512	TV		45	25.9725	415.9900	boat1024	TV		40	27.3254	1880.2
		1	63	25.9135	16.32%			1	57	27.3394	12.81%
		2	51	25.8780	15.32%			2	46	27.3523	11.40%
		3	46	25.8724	15.82%			3	43	27.3495	11.74%
lena512	32	4	43	25.9261	16.41%	boat1024	32	4	42	27.3081	12.96%
		5	41	25.9022	17.86%			5	42	27.2982	14.28%
		6	41	25.8859	19.84%			6	43	27.3290	15.88%
		7	45	25.9778	24.71%			7	42	27.2660	18.54%
		8	46	25.9247	28.62%			8	42	27.3165	21.10%

5.2. **Sensitivity to the subdomain size and overlapping size.** After understanding some general behaviors of the algorithms, we next demonstrate the time-saving by DD and DDC, and also provide some guidelines on the choice of the subdomain and overlapping sizes.

To illustrate the impact of overlapping sizes, we tabulate in Table 1 the PSNR and CPU time of the classic TV by the lagged diffusivity fixed-point iteration and DD with subdomain size 32, but with different overlapping size δ. Here, the percentage of the CPU time is against TV, and likewise for other tables. We see that that the PSNR obtained by DD is not so sensitive to the overlapping size δ, while the computational time increases as δ increases, as expected. To have a good trade-off between convergence rate and quality of restoration, it is advisable to choose δ to be $2, 3$ or 4. It is essential to point out that the use of DD leads to a remarkable reduction of computational time in particular for images of large size. One also refers to Figure 6. the restored lena-512×512 image.

We further examine the impact of subdomain size to the overall performance of DD. For this purpose, we fix the overlapping size $\delta = 4$, but vary α and the subdomain size d. Once again, Table 2 indicates a significant gain in computation time. It also shows that a good choice of d is roughly $1/16$ of the given image size.

TABLE 2. Comparison of computational time of TV and DD for different subdomain sizes of lena-512×512 and boat-1024×1024 with $\epsilon = \beta = 10^{-4}$, $\sigma = 0.04$ and the PSNR depicted in Figure 3.

$d \setminus \alpha$	200	100	67	50	40	35	30	25	22	20
TV512	570.90	516.20	392.57	415.48	415.99	359.06	357.23	326.31	313.74	300.79
8	71.30%	48.02%	46.40%	37.85%	34.98%	38.99%	36.51%	36.47%	34.33%	37.50%
16	38.62%	24.32%	23.17%	20.02%	18.79%	22.80%	22.54%	24.29%	23.57%	25.73%
32	26.58%	17.36%	18.81%	17.02%	15.86%	21.76%	19.77%	20.58%	19.07%	19.27%
64	33.83%	22.95%	29.86%	24.26%	24.16%	27.71%	26.26%	25.42%	27.98%	26.85%
128	34.16%	34.67%	42.94%	37.61%	26.60%	38.24%	35.65%	38.38%	31.70%	30.96%
TV1024	3145.4	2197.3	2061.4	1884.0	1880.2	1935.5	1805.1	1768.9	1859.6	1736.3
8	45.85%	43.26%	37.78%	34.07%	31.68%	28.56%	27.82%	26.85%	24.40%	23.89%
16	22.37%	20.42%	17.47%	16.24%	15.88%	16.44%	17.41%	16.81%	15.67%	15.85%
32	16.17%	13.53%	13.17%	14.09%	13.74%	13.63%	13.87%	13.98%	12.47%	12.67%
64	16.45%	16.91%	17.05%	19.35%	18.95%	18.01%	19.87%	19.77%	17.79%	17.48%
128	25.52%	33.50%	26.47%	30.09%	29.74%	29.23%	26.81%	25.02%	24.86%	26.00%

We now particularly examine the effect of coarse mesh correction. We tabulate in Table 3 the computational time and PSNR for three methods with various choice of subdomain sizes. We also test them on large size image cow-2048×2048. Once more, it indicates the notable time saving by using domain decomposition techniques, and also justifies the $1/16-$rule for the choice of subdomain size. However, we realize that the coarse mesh correction does not help too much to DD in the sense that almost the same number of iterations are needed for DD and DDC, but more time is consumed, since additional coarse mesh equations have to be resolved at each iteration (cf. Algorithm DDC). We believe that the main reason is the regularization constants α, β are small, which results in stiff elliptic equations and the correcting effect is expected to be minor. The influence of the coarse mesh correction increases when β is bigger as shown in Table 4.

We illustrate below some samples of the restored image obtained by DD or DDC with an "optimal" choice of the intrinsic parameters. Figure 6 is devoted to the

TABLE 3. Comparison of TV, DD and DDC for boat-1024×1024 and cow-2048×2048 with $\sigma = 0.04$, $\beta = \epsilon = 10^{-4}$, $\alpha = 40$ and various subdomain sizes. Here, PSNR1 and PSNR2 refer to the PSNR of DD and DDC, respectively, and likewise for the iteration number k and computational time T.

d	δ	PSNR1	PSNR2	$k1$	$k2$	$T1$	$T2$
TV1024			27.3254		40		1880.2
8	2	27.3786	27.3429	62	63	31.68%	57.72%
16	4	27.2813	27.3294	43	43	15.88%	33.67%
32	4	27.3089	27.3329	42	42	13.74%	29.32%
64	4	27.3031	27.2892	42	41	18.95%	31.66%
128	4	27.3066	27.2647	40	41	29.74%	39.95%
TV2048			25.3433		41		12319.0
8	2	25.8873	24.4947	79	79	26.47%	48.80%
16	4	25.4409	25.3809	47	46	11.13%	24.06%
32	4	25.3787	25.3675	43	42	8.25%	20.99%
64	4	25.3650	25.3499	42	42	12.57%	21.74%
128	4	25.3808	25.3956	41	41	15.23%	23.94%

TABLE 4. Comparison of TV, DD and DDC with $\beta = 0.1, \sigma = 0.04, \alpha = 40, \epsilon = 10^{-6}$.

d	δ	PSNR1	PSNR2	$k1$	$k2$	$T1$	$T2$
TV512			26.0541		163		1420.2
32	4	25.9820	25.9900	173	157	17.93%	31.44%
TV1024			27.3402		189		8129.9
32	4	27.3954	27.3377	246	202	16.36%	32.01%
TV2048			25.4310		141		35901.0
32	4	25.4304	25.4146	157	145	9.54%	22.49%

lena-512×512, where we depict the difference images and find that three methods with the same stopping rule give indistinguishable restored images. Indeed, the PSNR are very close and the dynamic error $\|u - u^K\|_2/\|u\|_2$ (where u is the true image, and u^K is the restored image by TV, DD or DDC with K steps) are TV: 0.8029, DD: 0.8076 and DDC: 0.8060. Figure 7 illustrates the restored image of larger size by DD.

Now, we examine the methods for very small regularization parameter β, and for images with considerable higher noise level. We record in Table 5 the iteration number k, computational time, and PSNR for TV and DD with the regularization parameter $\beta = 10^{-12}$. In this case, the system is very stiff, so TV is extremely costly, but the stiffness can be significantly relaxed by breaking down the size of system, so DD is very fast. In a nutshell, DD is still very efficient for small β.

Finally, we illustrate in Figures 8-9, the restoration of boat-1024×1024 and cow-2048×2048 with higher noise level by DD. As before, the quality is essentially the same as that by TV, but recovered by much less CPU time.

5.3. **A comparison with dual approaches.** To further show the performance of the domain decomposition method, we compare it with the dual algorithms in [14, 10, 16], which have been proven to be very efficient for the ROF model.

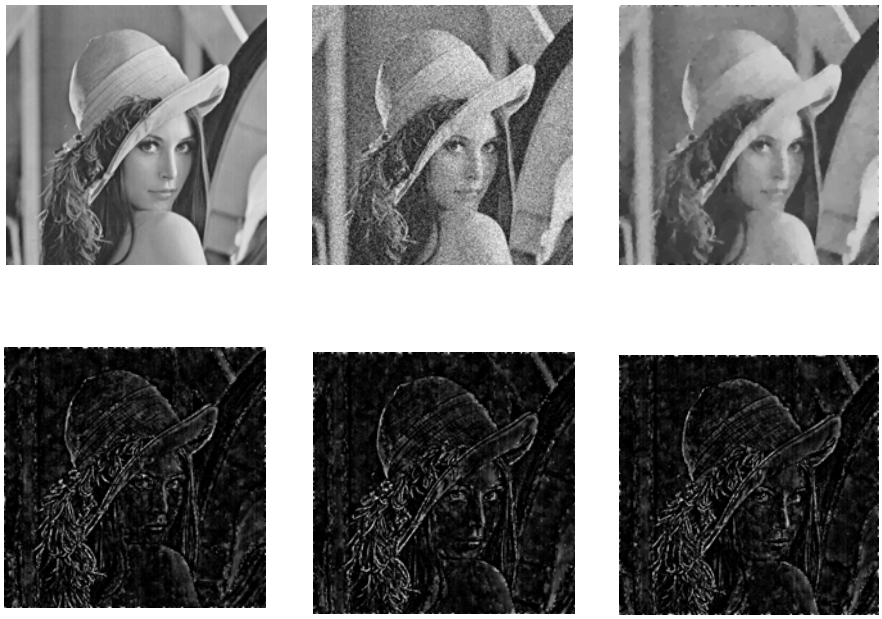


FIGURE 6. Row one: original image (left lena-512 × 512), noise image with $\sigma = 0.04$ (middle) and restored image(right) obtained by DD with subdomain size $d = 32$, overlapping size $\delta = 4$, $\alpha = 40$, $\epsilon = \beta = 10^{-4}$. Here, SNR= 9.0597 and PSNR= 25.9388. Row two: difference images between the restored one and original one (magnified by multiplying 5). From left to right: TV, DD, DDC with the same parameters chosen as above.

TABLE 5. Comparing TV with DD using small $\beta = 10^{-12}$, $\sigma = 0.04$, $\epsilon = 10^{-4}$ and $\alpha = 40$.

image	d	δ	PSNR	k	T
lena512	TV		26.0702	52	14873.0
lena512	8	2	26.0221	57	0.89%
	16	4	25.9995	53	0.65%
	32	4	25.9089	48	0.54%
	64	4	25.7169	54	1.02%
	128	4	25.6731	52	2.71%

We first briefly review the algorithm (cf. [16, p.199-200]). To solve (7), we introduce a new variable

$$\boldsymbol{p} = -\frac{\nabla u}{|\nabla u|_\beta}, \tag{29}$$

where $|\nabla u|_\beta = \sqrt{u_x^2 + u_y^2 + \beta}$ for some $\beta > 0$. For the ROF model, we obtain a system in $(u, \boldsymbol{p})$:

$$u = z - \alpha \mathrm{div}\boldsymbol{p}, \quad \nabla u + \boldsymbol{p}|\nabla u|_\beta = 0. \tag{30}$$

Eliminating u from the second equation leads to

$$-\boldsymbol{H} + \boldsymbol{p}\big|\boldsymbol{H}\big|_\beta = 0 \quad \text{with} \quad \boldsymbol{H}(\boldsymbol{p}) := -\nabla u/\alpha = \nabla\big(\mathrm{div}\boldsymbol{p} - z/\alpha\big). \tag{31}$$

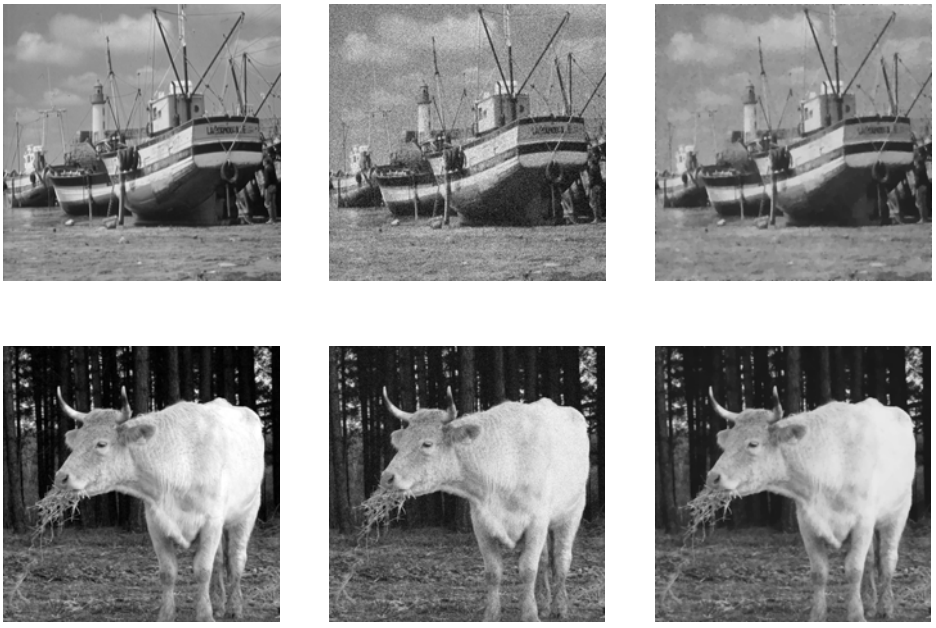


FIGURE 7. DD restoration of boat-1024×1024 and cow-2048×2048. Clean image (left), noise image with $\sigma = 0.04$ (middle) and restored image (right) with overlapping size $\delta = 4$, $\alpha = 40$, and $\epsilon = \beta = 10^{-4}$. Note: for boat, $d = 64$, SNR= 9.1503 and PSNR= 27.3031, and for cow, $d = 128$, SNR= 9.1866 and PSNR= 25.3808.

FIGURE 8. Original image boat-1024×1024 (left), noise image with $\sigma = 0.1$ (middle) and restored image(right) by DD with subdomain size $d = 64$, overlapping size $\delta = 4$, $\alpha = 67$, $\epsilon = 10^{-4}$ and $\beta = 10^{-12}$. Here, SNR= 6.0128, PSNR= 24.2600.

Following [10], we update the dual variable $\boldsymbol{p}$ by the scheme:

$$\boldsymbol{p}^0 = 0, \quad \boldsymbol{p}^{n+1} = \frac{\boldsymbol{p}^n + \tau \boldsymbol{H}(\boldsymbol{p}^n)}{1 + \tau |\boldsymbol{H}(\boldsymbol{p}^n)|_\beta}, \quad n = 0, 1, \cdots, \tag{32}$$

FIGURE 9. Original image cow-2048 × 2048 (left), noise image with $\sigma = 0.1$ (middle) and restored image(right) by DD with subdomain size $d = 128$, overlapping size $\delta = 4$, $\alpha = 67$, $\epsilon = 10^{-4}$ and $\beta = 10^{-4}$. Here, SNR= 5.8910 and PSNR= 21.6081.

where the gradient and divergence operators can be discretized by the forward and backward finite difference, respectively, together with the boundary condition $\boldsymbol{p}|_{\partial\Omega} = 0$, as in [10].

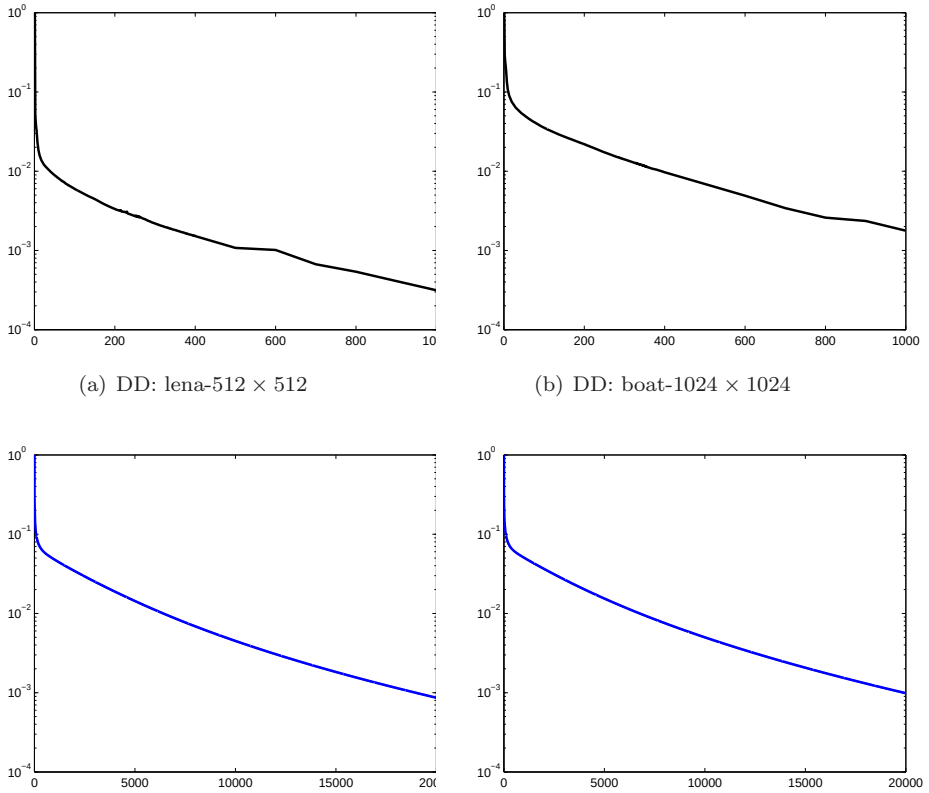
We compare the domain decomposition method with the dual algorithm in terms of decay of the residual

$$R^n := -\alpha \mathrm{div}\Big(\frac{\nabla u^n}{|\nabla u^n|_\beta}\Big) + (u^n - z), \quad n = 1, 2, \cdots. \tag{33}$$

Setting $u^n = z - \alpha \mathrm{div}\boldsymbol{p}^n$, we find from (30)-(32) that the residual of the dual algorithm can be computed by

$$R^n = \frac{\alpha^2}{\tau}\mathrm{div}\Big(\frac{\boldsymbol{p}^{n+1} - \boldsymbol{p}^n}{|\nabla u^n|_\beta}\Big) + (u^n - u^{n+1}), \quad n = 1, 2, \cdots. \tag{34}$$

In Figure 10, we plot the ratio $\|R^n\|_2/\|R^1\|_2$ (where $\|\cdot\|$ is the l^2-norm as before) versus the iteration number for both methods. In this comparison, we choose the noise level $\sigma = 0.02$, the parameters $\alpha = 40, \beta = 10^{-6}$, the time step $\tau = 1/16$ in (34), and the subdomain size $d = 32$ and overlapping size $\delta = 4$ in DD. We see from Figure 10 that the decay rate of the residual (33) of DD is more than 20 times faster than that of the dual algorithm (and the gain is even more for images of larger size). In other words, for a given accuracy, the DD with the current inner solver on each subdomain requires much smaller number of iterations. Hence, with this trade-off, the total computational cost of DD is comparable to that of the dual algorithm, in particular, for images of large size, although it is fairly sensitive to give a fair comparison. It is essential that the use of domain decomposition technique allows for breaking down the size of the problem, and one might choose a more efficient subdomain solver with a good balance between the accuracy and efficiency. We want to emphasis that our method can be used for general image processing problems. Here, we have used the well-known ROF to demonstrate that the domain decomposition approach can give superior performance in term of computing and memory efficiency.



(a) DD: lena-512 × 512 (b) DD: boat-1024 × 1024

(c) Dual: lena-512 × 512 (d) Dual: boat-1024 × 1024

FIGURE 10. Comparison of the decay of the l^2-norm of the residual (33) versus the iteration number for DD and the dual method. Here, the PSNRs at the final step for (a)-(d) are $23.0, 23.2, 23.5$ and 23.6, respectively.

Concluding Remarks. We presented in this paper a domain decomposition method for general image processing problem and we use the ROF model show its advantages. We described the very detailed implementation of the domain decomposition and coarse mesh correction techniques. We also provided a plenty of numerical results that demonstrated the effectiveness of the proposed methods in CPU time and memory saving. We gave useful guidelines for the choices of parameters through such quantitative studies. Moreover, we compared the proposed method with the dual algorithms in decay of numerical residuals. Most importantly, these techniques are very useful to decompose a large scale problem to some subproblems of much smaller size in an efficient manner. The proposed algorithm could be a viable approach for practical three dimensional data processing with large data sets.

REFERENCES

[1] R. Acar and C.R. Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse problems*, 10(6):1217–1230, 1994.

[2] G. Aubert and P. Kornprobst. *Mathematical problems in image processing: partial differential equations and the calculus of variations*. Springer-Verlag New York Inc, 2006.

[3] E. Bae and X.C. Tai. Graph cut optimization for the piecewise constant level set method applied to multiphase image segmentation. *Space Space and Variational Methods in Computer Vision, SSVM 2009*, pages 1–13, 2009.
[4] D.P. Bertsekas. Multiplier methods: a survey. *Automatica—J. IFAC*, 12(2):133–145, 1976.
[5] P. Blomgren, T.F. Chan, P. Mulet, L. Vese, and W.L. Wan. Variational PDE models and methods for image processing. *Numerical analysis*, 420:43–67, 1999.
[6] A.C. Bovik. *Handbook of image and video processing (communications, networking and multimedia)*. Academic Press, Inc. Orlando, FL, USA, 2005.
[7] J.H. Bramble, J.E. Pasciak, J. Wang, and J. Xu. Convergence estimates for product iterative methods with applications to domain decomposition. *Mathematics of Computation*, 56(193):1–21, 1991.
[8] M. Burger, G. Gilboa, S. Osher, and J. Xu. Nonlinear inverse scale space methods. *Commun. Math. Sci*, 4(1):179–212, 2006.
[9] M. Burger, S. Osher, J. Xu, and G. Gilboa. Nonlinear inverse scale space methods for image restoration. *Lecture Notes in Computer Science*, 3752:25–36, 2005.
[10] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, 2004.
[11] A. Chambolle. Total variation minimization and a class of binary MRF models. *Lecture Notes in Computer Science*, 3757:136–152, 2005.
[12] A. Chambolle and P.L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167–188, 1997.
[13] T.F. Chan, K. Chen, and X.C. Tai. Nonlinear multilevel schemes for solving the total variation image minimization problem. In *Image processing based on partial differential equations*, Math. Vis., pages 265–288. Springer, Berlin, 2007.
[14] T.F. Chan, G.H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 20(6):1964–1977, 1999.
[15] T.F. Chan and P. Mulet. Iterative methods for total variation image restoration. In *Iterative methods in scientific computing (Hong Kong, 1995)*, pages 359–381. Springer, Singapore, 1997.
[16] T.F. Chan and J. Shen. *Image processing and analysis*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005. Variational, PDE, wavelet, and stochastic methods.
[17] T.F. Chan and X.C. Tai. Identification of discontinuous coefficients in elliptic problems using total variation regularization. *SIAM Journal on Scientific Computing*, 25(3):881–904, 2004.
[18] Q.S. Chang and I.L. Chern. Acceleration methods for total variation-based image denoising. *SIAM Journal on Scientific Computing*, 25(3):982–994, 2004.
[19] K. Chen and X.C. Tai. A nonlinear multigrid method for total variation minimization from image restoration. *J. Sci. Comput.*, 33(2):115–138, 2007.
[20] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part I: Fast and exact optimization. *Journal of Mathematical Imaging and Vision*, 26(3):261–276, 2006.
[21] M. Dryja and O.B. Widlund. Towards a unified theory of domain decomposition algorithms for elliptic problems, Third International Symposiumon Domain Decomposition Methods for Partial Differential Equations, Houston, Texas, T. Chan et. al., eds, 1989.
[22] L.C. Evans and R.F. Gariepy. *Measure theory and fine properties of functions*. Studies in Advanced Mathematics. CRC Press, Boca Raton, FL, 1992.
[23] M. Fornasier, A. Langer, and C.B. Schönlieb. Domain decomposition methods for compressed sensing. *In print*, 2009.
[24] M. Fornasier, A. Langer, and C.B. Schönlieb. A convergent overlapping domain decomposition method for total variation minimization. *In print*, 2009.
[25] G. Gallo, M.D. Grigoriadis, and R.E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989.
[26] R. Glowinski and P. Le Tallec. *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*, volume 9. Society for Industrial Mathematics, Philadelphia, PA, 1989.
[27] D. Goldfarb and W. Yin. Parametric maximum flow algorithms for fast total variation minimization. *Rice University CAAM Technical Report TR07-09*, 2007.
[28] T. Goldstein and S. Osher. The split Bregman method for $L1$-regularized problems. *SIAM J. Imaging Sci.*, 2(2):323–343, 2009.

[29] M. Griebel and P. Oswald. On the abstract theory of additive and multiplicative Schwarz algorithms. *Numerische Mathematik*, 70(2):163–180, 1995.
[30] L. He, T.C. Chang, S. Osher, T. Fang, and P. Speier. MR image reconstruction by using the iterative refinement method and nonlinear inverse scale space methods. *UCLA CAM Report*, 6–35, 2006.
[31] R. Kimmel and I. Yavneh. An algebraic multigrid approach for image analysis. *SIAM Journal on Scientific Computing*, 24(4):1218–1231, 2003.
[32] T. Kohlberger, C. Schnorr, A. Bruhn, and J. Weickert. Domain decomposition for variational optical-flow computation. *IEEE Transactions on Image Processing*, 14(8):1125–1137, 2005.
[33] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics*, 22(3):277–286, 2003.
[34] T. Lu, P. Neittaanmaki, and X.C. Tai. A parallel splitting up method and its application to Navier-Stokes equations. *Applied Mathematics Letters*, 4(2):25–29, 1991.
[35] T. Lu, P. Neittaanmaki, and X.C. Tai. A parallel splitting up method for partial diferential equations and its application to Navier-Stokes equations. *RAIRO Mathematical Modelling and Numerical Analysis*, 26(6):673–708, 1992.
[36] M. Lysaker, S. Osher, and X.-C. Tai. Noise removal using smoothed normals and surface fitting. *IEEE Trans. Image Processing*, 13(10):1345–1357, 2004.
[37] F. Malgouyres. Minimizing the total variation under a general convex constraint for image restoration. *IEEE Trans. Image Process.*, 11(12):1450–1456, 2002.
[38] A. Marquina and S. Osher. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. *SIAM Journal on Scientific Computing*, 22(4):387–405, 2000.
[39] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling and Simulation*, 4(2):460–489, 2006.
[40] S. Osher and R.P. Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer, 2003.
[41] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*, 79:12–49, 1988.
[42] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.
[43] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.
[44] J. Savage and K. Chen. An improved and accelerated non-linear multigrid method for total-variation denoising. *International Journal of Computer Mathematics*, 82(8):1001–1015, 2005.
[45] B.F. Smith, P.E. Bjørstad, and W.D. Gropp. *Domain decomposition*. Cambridge University Press, Cambridge, 1996. Parallel multilevel methods for elliptic partial differential equations.
[46] K. Stüben. A review of algebraic multigrid. *J. Comput. Appl. Math.*, 128(1-2):281–309, 2001. Numerical analysis 2000, Vol. VII, Partial differential equations.
[47] X.C. Tai and Y. Duan. Domain decomposition methods with Graph cuts algotithms for image segmentation. *UCLA CAM Report*, 9–54, 2009.
[48] X.C. Tai and M. Espedal. Applications of a space decomposition method to linear and nonlinear elliptic problems. *Numerical Methods for Partial Differential Equations*, 14(6):717–737, 1998.
[49] X.C. Tai and M. Espedal. Rate of convergence of some space decomposition methods for linear and nonlinear problems. *SIAM Journal of Numerical Analysis*, 35(14):1558–1570, 1998.
[50] X.C. Tai and P. Tseng. Convergence rate analysis of an asynchronous space decomposition method for convex minimization. *Mathematics of Computation*, 71(239):1105–1136, 2002.
[51] X.C. Tai and C. Wu. Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model. *Space Space and Variational Methods in Computer Vision, SSVM 2009*, pages 502–513, 2009.
[52] X.C. Tai and J. Xu. Global and uniform convergence of subspace correction methods for some convex optimization problems. *Mathematics of Computation*, 71(237):105–124, 2002.
[53] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press Inc., San Diego, CA, 2001. With contributions by A. Brandt, P. Oswald and K. Stüben.
[54] 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(3):271–293, 2002.
[55] C.R. Vogel. *Computational methods for inverse problems*. Society for Industrial Mathematics, 2002.

[56] C.R. Vogel and M.E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.
[57] C.R. Vogel, M.E. Oman, et al. Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Transactions on Image Processing*, 7(6):813–824, 1998.
[58] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.
[59] Y. Wang, W. Yin, and Y. Zhang. A fast fixed-point algorithm for convex total variation regularization. Technical report, working paper, Rice University, Houston, TX, 2007.
[60] Y. Wang, W. Yin, and Y. Zhang. A fast algorithm for image deblurring with total variation regularization. *Rice University CAAM Technical Report TR07-10*, 2007.
[61] J. Weickert. Recursive separable schemes for nonlinear diffusion filters. *Lecture Notes in Computer Science*, 1252:260–271, 1997.
[62] J. Weickert, B. Romeny, and M.A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, 1998.
[63] Y.W. Wen, M.K. Ng, and Y.M. Huang. Efficient total variation minimization methods for color image restoration. *IEEE Transactions on Image Processing*, 17(11):2081–2088, 2008.
[64] J.C. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, 34(4):581–613, 1992.
[65] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for compressend sensing and related problems. *SIAM J. Imaging Sciences*, 1(1):143–168, 2008.

Received xxxx 20xx; revised xxxx 20xx.

E-mail address: jingxu@mail.zjgsu.edu.cn
E-mail address: tai@math.uib.no
E-mail address: lilian@ntu.edu.sg