DOMAIN DECOMPOSITION METHOD FOR IMAGE DEBLURRING

JING XU[1] AND HUI BIN, CHANG[2] AND JING QIN[3]

Abstract. As a fundamental problem in image processing, image deblurring has still attracted a lot of research attention. Due to the large computational cost, especially for high-resolution images, it becomes challenging to solve the deblurring minimization problem and the underlying partial differential equations. The domain decomposition method (DD), as one of the most efficient algorithms for solving large scale problems, had not been applied directly to image deblurring because of the global characteristic of the blur operator. In this paper, in order to avoid separating the blur operator, we propose an algorithm for directly solving the total variational based minimization problems with DD. Various numerical experiments and comparisons demonstrate that the larger the image size is, the more efficient the proposed method is in saving running time. The parallelization has also been realized by using the parallel computing toolbox of MATLAB.

1. Introduction

Image deblurring is a fundamental problem in both image processing and computer vision with broad applications. Given a blurry and noisy image $z : \Omega \to \mathbb{R}$,

$$z = Ku + n \tag{1.1}$$

where Ω is a bounded open set in $\mathbb{R}^2$, u is the underlying clean image, K, also called the point spread function (PSF), is a blur operator with $Ku = h * u$ where h is a convolution kernel with compact support (e.g., discrete Gaussian kernel), and n is a Gaussian white noise with zero mean. We aim to recover the unknown u and K from z.

Given the knowledge of the blur operator K, one of the most popular methods for noise removal and deblurring is the total variation based restoration method proposed by L. Rudin, S. Osher and E. Fatemi, where the total variation of u is used as a penalty functional in [15]. The corresponding image restoration can then be formulated as the following unconstrained minimization problem:

$$\min_u \left(\int_\Omega |\nabla u| \; dxdy + \frac{\lambda}{2} \|Ku - z\|_{L^2}^2 \right). \tag{1.2}$$

Here, $\lambda > 0$ is the penalty parameter. The functional in (1.2) is strictly convex with a unique global minimizer. If the convolution kernel h is the delta function δ satisfying $\delta * u = u$, we have $K = I$ and (1.2) is simply the original TV based image denoising model which restores the image from the noisy observation while preserving edges. For a general convolution kernel h, it becomes more

1991 *Mathematics Subject Classification.* 68U10, 65M55, 74S20.

Key words and phrases. Overlapping domain decomposition, Total variation minimization, image deblurring.

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 NSFC with Grant No. 11001239, No. 11101365, No. 11201420. The second author is partially supported by PHD Programme 52XB1304 of Tianjin Normal University and NSFC with Grant No. 11071080.

[1]School of Statistics and Mathematics, Zhejiang Gongshang University, 310018, P.R.China. Corresponding author, Email: jingxu@amss.ac.cn.

[2]School of Mathematical Sciences, Tianjin Normal University, 300387, P.R. China

[3]Department of Mathematics, University of California, Los Angeles, CA 90095, USA.

difficult to restore the image u and may return a solution which is sensitive to the perturbation of the input data due to the ill-posedness of the problem.

There is a considerable amount of work to solve the model, such as the gradient descent method [15], the dual model [6], the Bregman iteration [27], the augmented Lagrangian method [24] and multigrid methods [4] and so on. Xu et al. [25] provided a brief review of the aforementioned algorithms. The purpose of this paper is to propose a fast algorithm based on the overlapping domain decomposition technique to solve the TV based deblurring model (1.2). It is well known that domain decomposition methods are powerful iterative methods for solving partial differential equations [3, 9, 13, 17, 26]. Some recent progress has shown that DD are also efficient for some nonlinear elliptic problems and some convex minimization problems [20, 19, 21, 22] with mesh independent convergence. To the best of our knowledge, the domain decomposition methods have not been directly applied to the TV based deblurring problem so far. Some recent efforts have been devoted to study this problems [18, 14, 11, 10, 12, 7]. In Xu et al. [25], they have used the overlapping DD to image denoising which divided the original problem into subproblems over subdomain. However, as a global operator, the convolution operator K brings up obstacles to directly apply DD into the image deblurring.

In this paper, we propose a DD based image deblurring algorithm which combines the subspace correlation method and the lagged diffusivity fixed-point iteration. Following the idea proposed in [8, 25], we use the lagged diffusivity fixed-point iteration by moving the blur operator to the right hand side and adding a term on both sides to guarantee the convergence. Two methods are provided to handle the model (1.2). One is "linearization method" that uses the approximation value u^k at the k-th iteration to replace the term by $\nabla \cdot \left(\frac{\nabla u}{|\nabla u^k|} \right)$ to solve the subproblem at the $(k+1)$-th iteration. The other is to use "augmented Lagrangian method" (AML) by introducing the new variable $p = \nabla u$ and the Lagrangian multiplier μ.

By decomposing the image domain into overlapping subdomains, the original minimization problem related to the model (1.2) is reduced to a sequence of sub-minimization problems on the subdomains. If the sub-minimization problems are solved exactly, then the convergence of the generated sequence is trivial. Due to the degeneracy of the nonlinear equation associated to (1.2) involving the blur operator, it is difficult to obtain the convergence rate for the numerical solutions which will be studied further. Numerical experiments show its capability in processing images of large size and saving CPU time. The proposed method also has good potentials in solving large-scale problems which are feasible for parallel computing. Furthermore, the speed-up efficiency can be enhanced by more than 0.5.

The rest of the paper is organized as follows. In Section 2, we briefly review the domain decomposition algorithm in a general framework of the subspace correction method. The finite-difference discretization schemes and the details of the algorithm are shown in Section 3. Various numerical experiments and discussions are shown to demonstrate the merits of the proposed methods in Section 4. In Section 5, we make the conclusions.

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

$$\min_{u \in V} F(u). \tag{2.1}$$

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

$$V = V_1 + V_2 + \cdots + V_m, \tag{2.2}$$

which implies that

$$\sum_{j=1}^{m} v_j \in V, \quad \forall\ v_j \in V_j,$$

and for any $v \in V$ there exists $v_j \in V_j$ such that $v = \sum_{j=1}^{m} v_j$. Following the framework of [26] for linear problems, we solve a sequence of sub-minimization problems over the subspaces:

$$\min_{e \in V_j} F(u^n + e), \quad j = 1, 2, \ldots, m, \tag{2.3}$$

where u^n denotes the n-th approximation to resolve (2.1). Two types of subspace correction methods based on (2.2)-(2.3), known as the parallel subspace correction (PSC) and successive subspace correction (SSC) method, were proposed in [26, 22]. Here, we adopt the latter that can also be parallelized by coloring techniques.

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

$$\Omega = \bigcup_{j=1}^{m} \Omega_j, \quad \Omega_j \cap \Omega_k \neq \emptyset, \quad k \neq j. \tag{2.4}$$

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

$$M := \sum_{j=1}^{m} m_j. \tag{2.5}$$

In Figure 2.1, we illustrate schematically the decomposition of Ω into four colored subdomains with 25 blocks. Based on the above decomposition scheme, we decompose the space $V = BV(\Omega)$ as

$$V = \sum_{j=1}^{m} V_j, \quad V_j = BV_0(\Omega_j), \tag{2.6}$$

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-deblurring model leads to an iterative algorithm. In the following, we give a detailed description of the two proposed algorithms.

2.1. **Algorithm I (Linearization Method):** First, we apply the DD method to the deblurring model (1.2) directly, namely (2.1) with

$$F(u) = \int_{\Omega} |\nabla u|\ dxdy + \frac{\lambda}{2}\|Ku - z\|_{L^2}^2.$$

The corresponding Euler-Lagrange equation is:

$$-\operatorname{div}\left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}\right) + \lambda K^*(Ku - z) = 0. \tag{2.7}$$

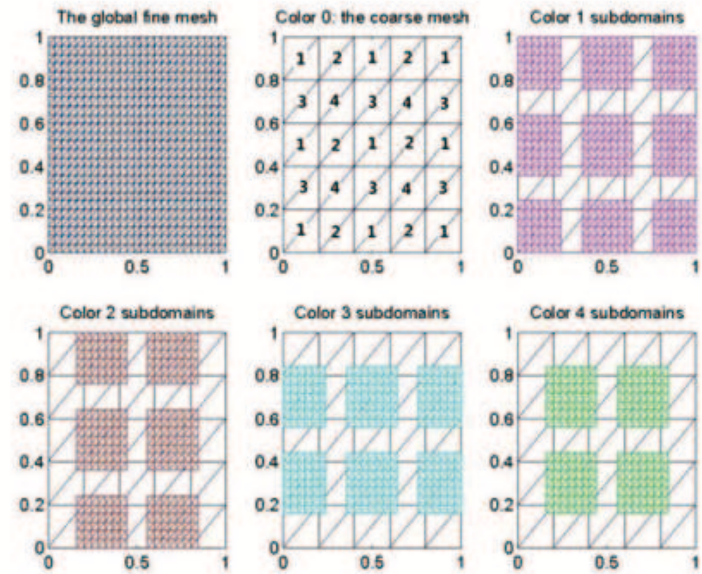

FIGURE 2.1. Schematic illustration of the coloring of the subdomains, and fine/coarse meshes on $\Omega = (0,1)^2$. This corresponds to the decomposition: $V^h = V_0^H + \sum_{i=1}^{4} V_i^h$ with $H = 5h$ and in (2.5), $m = 4$, $m_1 = 9, m_2 = 6, m_3 = 6, m_4 = 4$, and $M = 25$.

Here, to avoid dividing by zero, we introduce a positive small number β in the denominator of the diffusion term. The differential equation (2.7) has been proven well-posed as $\beta \to 0^+$ in [1] . Instead of truncating the global blur operator K, we rewrite (2.7) as follows

$$-\operatorname{div}\left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}\right) = \lambda K^*(z - Ku). \tag{2.8}$$

For some blur operators, the iterative numerical methods to solve the above differential equation will diverge. Adding one stable term $\mathcal{B}u$ (c.f. [8]) to both sides of the equations will resolve this issue. We choose the algorithm in our numerical experiment as follows:

$$\mathcal{B}u - \operatorname{div}\left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}\right) = \lambda K^*(z - Ku) + \mathcal{B}u. \tag{2.9}$$

with a homogenous Neumann boundary condition $\partial u/\partial \mathbf{n} = 0$. We will discuss how to choose a proper $\mathcal{B}$ in Section 4. Once the convergence of the algorithm is ensured, we are able to split the entire image into small rectangles and solve efficiently the corresponding boundary value problems on each rectangle. Recall that the lagged diffusivity fixed-point iteration (cf. [23]) for (2.9) is to solve the linearized equation

$$\mathcal{B}u^{k+1} - \operatorname{div}\left(\frac{\nabla u^{k+1}}{\sqrt{|\nabla u^k|^2 + \beta}}\right) = \lambda K^*(z - Ku^k) + \mathcal{B}u^k, \quad k = 0, 1, \cdots, \tag{2.10}$$

with the initial value u^0. Since each iteration involves all the pixel intensities in the image domain, it will be computationally intensive and usually cause the ill-conditioning of the system when the image size is large. The domain decomposition based SSC algorithm will overcome the difficulties by decomposing the whole problem into sub-problems on much smaller subdomains.

Given an initial value $u^0 \in V$, the SSC algorithm generates u^{n+1} by

$$\begin{cases} F\Big(u^{n+\frac{j-1}{m}} + e_j^n\Big) \leq F\Big(u^{n+\frac{j-1}{m}} + v_j\Big), \quad \forall\ v_j \in V_j = BV_0(\Omega_j), \\ u^{n+\frac{j}{m}} = u^{n+\frac{j-1}{m}} + e_j^n, \quad 1 \leq j \leq m. \end{cases} \tag{2.11}$$

Notice that e_j^n is the solution of the subproblem over Ω_j. Therefore, each boundary value problem over Ω_j has the following form

$$\begin{cases} \mathcal{B}u^{n+\frac{j}{m}} - \operatorname{div}\left(\frac{\nabla u^{n+\frac{j}{m}}}{\sqrt{|\nabla u^{n+\frac{j-1}{m}}|^2+\beta}}\right) = \lambda K^*(z - Ku^{n+\frac{j-1}{m}}) + \mathcal{B}u^{n+\frac{j-1}{m}}, & \text{in } \Omega_j, \\ \dfrac{\partial u^{n+\frac{j}{m}}}{\partial \boldsymbol{n}} = 0, & \text{on } \partial\Omega_j \cap \partial\Omega, \\ u^{n+\frac{j}{m}} = u^{n+\frac{j-1}{m}}, & \text{on } \partial\Omega_j \setminus \partial\Omega. \end{cases} \tag{2.12}$$

One can see that $u^{n+\frac{j}{m}} = u^{n+\frac{j-1}{m}}$ for $x \in \Omega \backslash \Omega_j$. The above iterative algorithm requires us to solve a sequence of minimization problems over the subspaces/subdomains.

2.2. **Algorithm II (Augmented Lagrangian Method).** Second, we try to solve the model (1.2) using the Augmented Lagrangian Methods (ALM) which is one of the most efficient algorithms. In [24], the authors applied the ALM method to solve (1.2) and showed that the dual method and the split Bregman iteration can actually be either deduced from, or equivalent to the ALM method. They both are just different iterative strategies to solve the same system resulted from a Lagrangian and penalty approach. In fact, ALM can be replaced by other fast numerical methods to solve subproblems which implies that our technique can be easily combined with other methods.

Instead of directly solving the Euler-Lagrange equation of (1.2) using the ALM method [25], one solves the constrained optimization problem by

$$\min_{u,q} \max_{\mu} L_{ROF} = \int_\Omega |q| + \frac{\lambda}{2}\|Ku - z\|^2 + \int_\Omega \mu \cdot (q - \nabla u) + \frac{r}{2}\int_\Omega |q - \nabla u|^2, \tag{2.13}$$

where $\mu = (\mu_1, \mu_2)^T$ is the Lagrange multiplier and r is a positive constant. Then the method is to seek a saddle point of the augmented Lagrangian functional $L_{ROF}(u, q, \mu)$.

To solve the problem of (2.13), we split it into the following two sub-problems [25]:

$$\arg\min_u F(u) = \frac{\lambda}{2}\|Ku - z\|^2 - \int_\Omega \mu \cdot \nabla u + \frac{r}{2}\int_\Omega |q - \nabla u|^2, \tag{2.14}$$

for a given q. Here $F(u)$ is the same as that in the equation (2.1), which is the second example of applying the domain decomposition method to (1.2) and

$$\arg\min_q \int_\Omega |q| + \int_\Omega \mu \cdot q + \frac{r}{2}\int_\Omega |q - \nabla u|^2, \tag{2.15}$$

for a given u. The sub-problems (2.14) and (2.15) can be efficiently solved. For (2.14), Euler-Lagrange equation is

$$\lambda K^*(Ku - z) + \nabla \cdot \mu + r\nabla \cdot q - r\triangle u = 0.$$

We use the same idea as in solving equation (2.7). The blur term and the term without u is moved to the right hand

$$-r\triangle u = \lambda K^*(z - Ku) - \nabla \cdot \mu - r\nabla \cdot q.$$

Then we add a $\mathcal{B}u$ to the both sides of the equation for convergence,

$$\mathcal{B}u - r\triangle u = \mathcal{B}u + \lambda K^*(z - Ku) - \nabla \cdot \mu - r\nabla \cdot q.$$

The lagged diffusivity fixed-point iteration (cf. [23]) is used to update u

$$\mathcal{B}u^{n+1} - r\triangle u^{n+1} = \mathcal{B}u^n + \lambda K^*(z - Ku^n) - \nabla \cdot \mu^n - r\nabla \cdot q^n.$$

After applying the SSC algorithm to the ALM for the given μ^n and q^n, we obtain $u^{n+\frac{j}{m}}$ by solving the following boundary value problem:

$$\begin{cases} \mathcal{B}u^{n+\frac{j}{m}} - r\triangle u^{n+\frac{j}{m}} = \mathcal{B}u^{n+\frac{j-1}{m}} + \lambda K^*(z - Ku^{n+\frac{j-1}{m}}) - \nabla \cdot \mu^n - r\nabla \cdot q^n. & \text{in } \Omega_j, \\ \dfrac{\partial u^{n+\frac{j}{m}}}{\partial \boldsymbol{n}} = 0, & \text{on } \partial\Omega_j \cap \partial\Omega, \\ u^{n+\frac{j}{m}} = u^{n+\frac{j-1}{m}}, & \text{on } \partial\Omega_j \setminus \partial\Omega. \end{cases} \tag{2.16}$$

We reformulate the problem (2.15) to be

$$\arg\min_q \int_\Omega |rq| + \frac{1}{2}\int_\Omega |rq - (r\nabla u - \mu)|^2.$$

Then

$$q = \frac{1}{r}\mathrm{prox}_{|\cdot|}(\omega) = \frac{1}{r}\max\{|\omega| - 1, 0\}\mathrm{sign}(\omega),$$

where $\omega = r\nabla u - \mu$ in [5]. In the discrete setting, we have

$$q^{n+1} = \begin{cases} \frac{1}{r}(1 - \frac{1}{|r\nabla u^{n+1} - \mu^n|})(r\nabla u^{n+1} - \mu^n) & |r\nabla u^{n+1} - \mu^n| > 1 \\ 0 & |r\nabla u^{n+1} - \mu^n| \leq 1 \end{cases} \tag{2.17}$$

Finally, we updated μ by

$$\mu^{n+1} = \mu^n + r(q^{n+1} - \nabla u^{n+1}).$$

3. Numerical discrete algorithm for TV deblurring

Based on our experience, the coarse grid correction does not help much for the TV-denoising in [25]. As such, we just present the one-level algorithm described in the previous section for the TV-deblurring model.

In order to solve (2.9) numerically, we first partition the domain $\Omega = (0, 1) \times (0, 1)$ into $L \times L$ uniform cells with mesh size $h = 1/L$, whose centers are

$$(x_l, y_k) = \left(\left(l - \frac{1}{2}\right)h, \left(k - \frac{1}{2}\right)h\right), \qquad l, k = 1, \cdots, L.$$

By applying the standard five-point stencil to the Laplacian operator, we get

$$\mathcal{B}u - \nabla_h \cdot \left(\frac{\nabla_h u}{\sqrt{|\nabla_h u|^2 + \beta}}\right) = \mathcal{B}u + \lambda K^*(z - Ku), \tag{3.1}$$

or more precisely,

$$
\begin{aligned}
&\mathcal{B}u_{l,k} - \frac{1}{h}\left(a_{l+\frac{1}{2},k}\frac{u_{l+1,k}-u_{l,k}}{h} - a_{l-\frac{1}{2},k}\frac{u_{l,k}-u_{l-1,k}}{h}\right)\\
&-\frac{1}{h}\left(a_{l,k+\frac{1}{2}}\frac{u_{l,k+1}-u_{l,k}}{h} - a_{l,k-\frac{1}{2}}\frac{u_{l,k}-u_{l,k-1}}{h}\right)\\
&= \mathcal{B}u_{l,k} + \left(\lambda K^*(z-Ku)\right)_{l,k}, \qquad\qquad l,k = 1,\cdots,L,
\end{aligned}
\tag{3.2}
$$

with discrete homogeneous Neumann boundary condition by one-sided second-order finite differences when $x=0$:

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

where

$$a_{l+\frac{1}{2},k} = \frac{1}{\sqrt{\left((D_x u)_{l+\frac{1}{2},k}\right)^2 + \left((D_y u)_{l+\frac{1}{2},k}\right)^2 + \beta}},$$

with

$$(D_x u)_{l+\frac{1}{2},k} = \frac{u_{l+1,k}-u_{l,k}}{h}, \quad (D_y u)_{l+\frac{1}{2},k} = \frac{1}{2}\left(\frac{u_{l+1,k+1}-u_{l+1,k-1}}{2h} + \frac{u_{l,k+1}-u_{l,k-1}}{2h}\right)$$

etc. To simplify the notation, we abbreviate (3.1) as

$$\mathcal{B}u + L(u)u = \mathcal{B}u + \lambda K^*(z-Ku), \tag{3.3}$$

where

$$L(v)w = -\nabla_h \cdot \left(\frac{\nabla_h w}{\sqrt{|\nabla_h v|^2+\beta}}\right). \tag{3.4}$$

In (3.3), $L(u)$ is fully nonlinear with widely varying coefficients. Moreover, the matrix K^*K is wide-banded and the spectrums of the matrices $L(u)$ and K^*K are quite differently distributed. We list the algorithm for the TV-deblurring model in the whole domain without DD in Algorithm I.

Algorithm I: TV-Deblurring.

1. Start with $u^0 = z$.
2. Given u^n, solve for u^{n+1} by (iterating on n):

$$\left(\mathcal{B} + L(u^n)\right)u^{n+1} = -(\lambda K^*K - \mathcal{B})u^n + \lambda K^* z$$

   Let

$$
\begin{aligned}
A &= \mathcal{B} + L(u^n)\\
F &= -\left(\lambda K^*K - \mathcal{B}\right)u^n + \lambda K^* z,
\end{aligned}
$$

   Then

$$AU = F$$

3. Go to next iteration for n.

Then the more detailed discrete forms of (2.12) are as follows:

$$
\begin{aligned}
&\mathcal{B}u_{l,k}^{n+\frac{j}{m}} - \left\{\delta_x^- \left[\frac{\delta_x^+ u_{l,k}^{n+\frac{j}{m}}}{\sqrt{(\delta_x^+ u_{l,k}^{n+\frac{j-1}{m}})^2 + (\delta_y^+ u_{l,k}^{n+\frac{j-1}{m}})^2 + \beta_h}}\right]\right.\\
&\qquad \left. + \delta_y^- \left[\frac{\delta_y^+ u_{l,k}^{n+\frac{j}{m}}}{\sqrt{(\delta_x^+ u_{l,k}^{n+\frac{j-1}{m}})^2 + (\delta_y^+ u_{l,k}^{n+\frac{j-1}{m}})^2 + \beta_h}}\right]\right\} = F^{n+\frac{j-1}{m}},
\end{aligned}
\tag{3.5}
$$

where $\delta_x^+, \delta_x^-, \delta_y^+, \delta_y^-$ denote the backward and forward difference scheme in the common sense which are the abbreviated forms of (3.2).

Compared with TV Deblurring, for which there are no nonlinear terms in the ALM equation, it is easy to discretize the Laplace operator for solving u with the finite difference discretization scheme. We provide the algorithm without DD as below:

Algorithm II: TV-ALM Deblurring.
Start with $u^0 = z$, $q^0 = (0,0)^T$, $\mu^0 = 0$
Assume we have u^n, q^n, μ^n
Solve for u^{n+1} by (iterating on n):

$$(\mathcal{B} - r\triangle)\, u^{n+1} = -(\lambda K^*K - \mathcal{B})u^n + \lambda K^* z - r\nabla \cdot q^n$$

Let

$$\begin{array}{rcl} A & = & \mathcal{B} - r\triangle \\ F & = & -\left(\lambda K^*K - \mathcal{B}\right) u^n + \lambda K^* z - \nabla \cdot \mu^n - r\nabla \cdot q^n, \end{array}$$

Then

$$AU = F$$

Solve for q^{n+1} by (iterating on n):

$$q = \begin{cases} \frac{1}{r}\left(1 - \frac{1}{|r\nabla u^{n+1} - \mu^n|}\right)(r\nabla u^{n+1} - \mu^n) & |r\nabla u^{n+1} - \mu^n| > 1 \\ 0 & |r\nabla u^{n+1} - \mu^n| \leq 1 \end{cases}$$

Solve for μ^{n+1} by (iterating on n):

$$\mu^{n+1} = \mu^n + r(q^{n+1} - \nabla u^{n+1}).$$

Go to the next iteration for n.

Similar to equation (3.5), obtained by applying the ALM method, we have

$$(\mathcal{B} - r(\delta_x^- \delta_x^+ + \delta_y^- \delta_y^+))u_{l,k}^{n+\frac{j}{m}} = F^{n+\frac{j-1}{m}}. \tag{3.6}$$

We list below a few possible choices for B (c.f. [8]) with identity matrix I:

(a)

$$\mathcal{B} = bI, \qquad b > b^* = \frac{\lambda}{2} \max_j \sum_i (K^*K)_{i,j} \tag{3.7}$$

(b)

$$\mathcal{B} = \mathrm{diag}(\lambda K^*K) + \gamma I, \qquad \gamma > \gamma^* = \frac{\lambda}{2} \max_i \left(\sum_{j \neq i} (K^*K)_{i,j} - (K^*K)_{i,j} \right) \tag{3.8}$$

(c)

$$\mathcal{B} = \frac{\lambda}{2}\mathrm{diag}(K^*K) + \frac{1}{2}\delta I, \qquad \delta > \delta^* = \lambda \max_i \sum_{j \neq i} (K^*K)_{i,j} \tag{3.9}$$

In this paper, we have adopted (3.7) for simplification. Both (3.8) and (3.9) have been proven experimentally to perform equivalently. The convergence analysis can be traced to [8] and [16].

Next we state the two algorithm for sub-domains proposed in Section 2.

3.1. **Algorithm I_{DD}: DD-Linearization Method.** In the following, we provide the TV deblurring's DD forms of Algorithm I in the above section.

Algorithm I_{DD}: TV DD-Deblurring.
Choose an initial value $u_h^0 \in V^h$.

For $n = 0$,

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

Compute F^n

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

Solve (3.5): $A_{h,j}\tilde{u}_h^{n+j/m} = F^{n+\frac{j-1}{m}}$.

end

Go to next iteration for n.

3.2. **Algorithm II_{DD}: DD-Augmented Lagrangian Method.** As a popular and efficient solver for the subproblem, we present the augmented Lagrangian method in Subsection 2.2 where $\mathcal{B}$ is chosen the same as B in Algorithm II.

Algorithm II_{DD}: TV ALM DD-Deblurring.
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 (3.6) for updating u, $A_{h,j}\tilde{u}_h^{n+j/m} = F^{n+\frac{j-1}{m}}$.

end

update q^n and μ^n,
Go to next iteration for n.

4. Numerical results

Next we present various numerical results to demonstrate the efficiency of the proposed domain decomposition based image deblurring algorithms.

We first give an example to show the general structure of a blur operator. If a blur operator given by the unit kernel (support size=5×5) as

$$\frac{1}{64}(1,1,4,1,1)^T(1,1,4,1,1) = \frac{1}{64}\begin{bmatrix} 1 & 1 & 4 & 1 & 1 \\ 1 & 1 & 4 & 1 & 1 \\ 4 & 4 & 16 & 4 & 4 \\ 1 & 1 & 4 & 1 & 1 \\ 1 & 1 & 4 & 1 & 1 \end{bmatrix},$$

then we let

$$T_2 = \begin{bmatrix} 4 & 1 & 1 & & & & 0 \\ 1 & 4 & 1 & 1 & & & \\ 1 & 1 & 4 & 1 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & 1 & 4 & 1 & 1 \\ & & & 1 & 4 & 1 & 1 \\ 0 & & & & 4 & 1 & 1 \end{bmatrix}_{L*L},$$

and

$$T_0 = 4T_2, \qquad T_1 = T_2.$$

Therefore, the blur matrix can be written as

$$K = \frac{1}{64} \begin{bmatrix} T_0 & T_1 & T_2 & & & & 0 \\ T_1 & T_0 & T_1 & T_2 & & & \\ T_2 & T_1 & T_0 & T_1 & T_2 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & T_2 & T_1 & T_0 & T_1 & T_2 \\ & & & T_2 & T_1 & T_0 & T_1 \\ 0 & & & & T_2 & T_1 & T_0 \end{bmatrix}_{L^2*L^2}.$$

Moreover, if a blur kernel can be expressed as

$$H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ h_{n1} & h_{n2} & \cdots & h_{nn} \end{bmatrix}_{n*n},$$

then the maximal diagonal entry of K^*K is $\sum_{i,j=1}^{n} h_{i,j}^2$ and the corresponding largest sum of off-diagonal entries of K^*K is $\sum_{i,j=1}^{n}\sum_{k,l=1,k\neq i,l\neq j}^{n} h_{i,j}h_{k,l}$. Based on the construction of K, we are able to choose a proper $\mathcal{B}$.

Here, we provide two types of blur operators used in our numerical experiments.

(1) A motion blur example with 90 degree (support size $=5\times 5$) given by the kernel

$$\frac{1}{5}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

(2) A truncated Gaussian blur given by the mask (support size$=5\times 5$)

$$h(x,y) = \begin{cases} ce^{-\tau(x^2+y^2)}, & \text{if } |x|,|y| \leq \frac{2}{L}, \\ 0, & \text{otherwise,} \end{cases}$$

Here the strength of the blur depends on parameters τ, c . Stronger blurs correspond to smaller values of τ or larger values of c. For example, the original images contain 256×256 pixels. Then $L = 256$, $|x|,|y| \leq \frac{1}{128}$ is equal to $|i| \leq 2$, $|j| \leq 2$ of the first blur operator.

We have

$$h * f(x,y) = \int_{-\frac{1}{128}}^{\frac{1}{128}} \int_{-\frac{1}{128}}^{\frac{1}{128}} ce^{-\tau((x-z)^2+(y-w)^2)} f(x,w)dzdw$$
$$= \int_{-2}^{2}\int_{-2}^{2} \frac{c}{L^2} e^{-\frac{\tau}{L^2}((X-Z)^2+(Y-W)^2)} f(\frac{Z}{L},\frac{W}{L})dZdW.$$

Then, we take $\tau' = \frac{\tau}{L^2}$ and

$$c' = \frac{\frac{c}{L^2}}{\sum_{i,j=-2}^{2} e^{-\tau'(i^2+j^2)}}.$$

Analogously, let

$$H = (e^{-2^2\tau'}, e^{-\tau'}, \quad 1, \quad e^{-\tau'}, e^{-2^2\tau'}),$$

then K can be considered as an operator defined by the kernel $c'H^TH$.

The performance is assessed by comparing with the lagged diffusivity fixed-point iteration (i.e., (2.10), denoted by TV) in terms of convergence, recovered image qualities measured in peak signal-to-noise ratio (PSNR) and the computational time. To the best of our knowledge, these algorithms have not been applied to the image deblurring problems so far. Hence, it is interesting to see some good results and particularly how the methods can be applied to restore images of large size.

By default, the pixel values of all images lie in the interval $[0, 255]$, and the Gaussian white noise is generated in MATLAB by using the command `imnoise(I,'gaussian',` M`,` σ`)` (i.e., the mean M and variance σ). In our tests, we use PSNR in [2] to measure the restored image quality, which is defined as in the logarithmic decibel scale:

$$\text{PSNR} = 10\log_{10} \frac{255^2}{\frac{1}{mn}\sum_{i,j}(u_{i,j} - z_{i,j})^2}, \tag{4.1}$$

where u is the restored image, and z is the original one. Typical values for the PSNR in lossy image and video compression are between 30dB and 50dB. The higher the PSNR, the better the quality of the image is. We use the relative error between two consecutive iterations as the stopping criterion:

$$\frac{\|u^k - u^{k-1}\|_2}{\|u^k\|_2} < \varepsilon, \tag{4.2}$$

where ε is a prescribed tolerance.

We test the methods on four images: shape-128×128, map-256×256, lena-512×512, boat-1024×1024 (See Figure 4.1). To simulate the noisy blurry observed images, we add the zero mean Gaussian noise with the standard deviation $\varepsilon = 10^{-4}$, and apply the Gaussian blur and the motion blur to the clean test images. Regarding the efficiency comparison, we compute the ratio of the elapsed time in the DD method to that in the TV method.

4.1. **Algorithm I TV-Deblurring v.s. Algorithm I_{DD} TV DD-Deblurring.** In Figure 4.2, we seek a good choice of the parameter λ in the TV and DD methods with different noise levels, where the left image uses $\sigma = 10^{-4}$, and the right one uses $\sigma = 10^{-3}$. From the result of Figure 4.2, one can see that $\lambda = 0.1$ is an ideal choice for two testing images. To make the comparison fair, we select the best restored image for each method in terms of PSNR. In Table 4.1, we look for a good overlapping size δ. Here we fix the subdomain size $d = 32$. One can see that the three supporting sizes yield the same results. In particular, $\delta = 1$ gives the best results no matter what the supporting size is. The running time of the DD method with $\delta = 1$ is less than 10% of that of the TV method while the corresponding PSNR is even higher than that of the TV method. The running time grows as the δ increases. All the running time of the DD mehod is less than 23% of the TV method. In addition, we test the different subdomain size under different blur kernel operators in Table 4.2 for the 512×512 lena image and Table 4.3 for the 1024×1024 lena image. We find the best choice is $d = 32$ for the running time, which is under 7% in the 512×512 lena image and 15% in the 1024×1024 boat image. Although it certainly takes more running time for both the DD method and the TV method for larger images, the time that the DD method saves actually grows as the image size increases based on Table 4.2 and Table 4.3. The results from the Gaussian blur is same as those for the motion blur. When dealing with the image of size 2048×2048 or even larger, the TV method is easily prone to fail due to the limited computer memory while the proposed DD method always works by shrinking the subdomains. More detailed results are shown in Table 4.1 to 4.3. Using the optimal λ, overlapping size, and subdomain size from the tests, Figure 4.3 and 4.4 illustrate the compared performance of Algorithm I and Algorithm I_{DD}. Here we also use Figure 4.5, 4.6 and 4.7 to show that the image features of the DD results are the same as them in the result obtained from the original TV model, while it can reduce the running time significantly.

4.2. **Algorithm II TV-ALM Deblurring v.s. Algorithm II_{DD} TV-ALM DD-Deblurring.** Under the same tests with different parameters, we test the 512×512 lena image and get the results directly from the server using **matlabpool** in jackfruit-sever with 4 Intel(R) Xeon(TM)MP CPU 3.33GHz, 16G RAM. The results are in Table 4.4.

We use **matlabpool** in the parallel computing toolbox of MATLAB. Our server owns four computing cores (P=4). Speed-up ratio refers to how much a parallel algorithm is faster than the corresponding sequential algorithm. We see that the speed-up ratio is about $2 \sim 3.5$ so the parallel efficiency $E(E = \frac{SP}{P})$ is about $0.5 \sim 0.87$. Parallel efficiency usually takes a value between zero and one, estimating how well-utilized the processors are in solving the problem, compared to how much effort is wasted in communication and synchronization. In the realization of parallel computing, we use one core for saving data and the other 3 left cores to compute the subproblem. Then the parallel efficiency is getting larger when setting $P = 3$, that is close to 1.

One can see that the proposed methods have been successfully applied to the image deblurring and lead to significant time and memory saving. Moreover, they are not sensitive to the image size.

TABLE 4.1. Different overlapping size δ using image lena512 with tolerance $\varepsilon = 10^{-4}$, $\sigma = 10^{-4}$, $\lambda = 0.1$ and $\beta = 10^{-5}$, sub-domain size $d = 32$, Gaussian blur with support size $S = 5^2, 7^2, 9^2$. We compute the ratio of the elapsed time in the DD method to that in the TV method.

S	δ	k	PSNR	Time	S	k	PSNR	Time	S	k	PSNR	Time
TV	0	124	37.8835	839.2698		131	38.2565	915.5855		135	38.3278	738.2279
	1	120	38.1716	8.09%		128	38.5557	8.02%		132	38.5601	9.97%
	2	124	37.8219	9.43%		130	38.2328	9.10%		137	38.0984	11.64%
5^2	3	124	37.7878	10.41%	7^2	130	38.2147	10.04%	9^2	138	38.0054	12.90%
	4	124	37.7392	11.96%		131	38.1424	11.62%		137	38.0591	14.62%
	5	124	37.7187	13.34%		131	38.1340	12.96%		139	37.9877	16.44%
	6	124	37.7318	14.48%		131	38.1503	14.12%		140	37.9686	18.05%
	7	125	37.7114	16.53%		132	38.0911	15.77%		141	37.9388	20.50%
	8	125	37.7175	18.02%		133	38.0659	17.09%		140	37.9323	22.30%

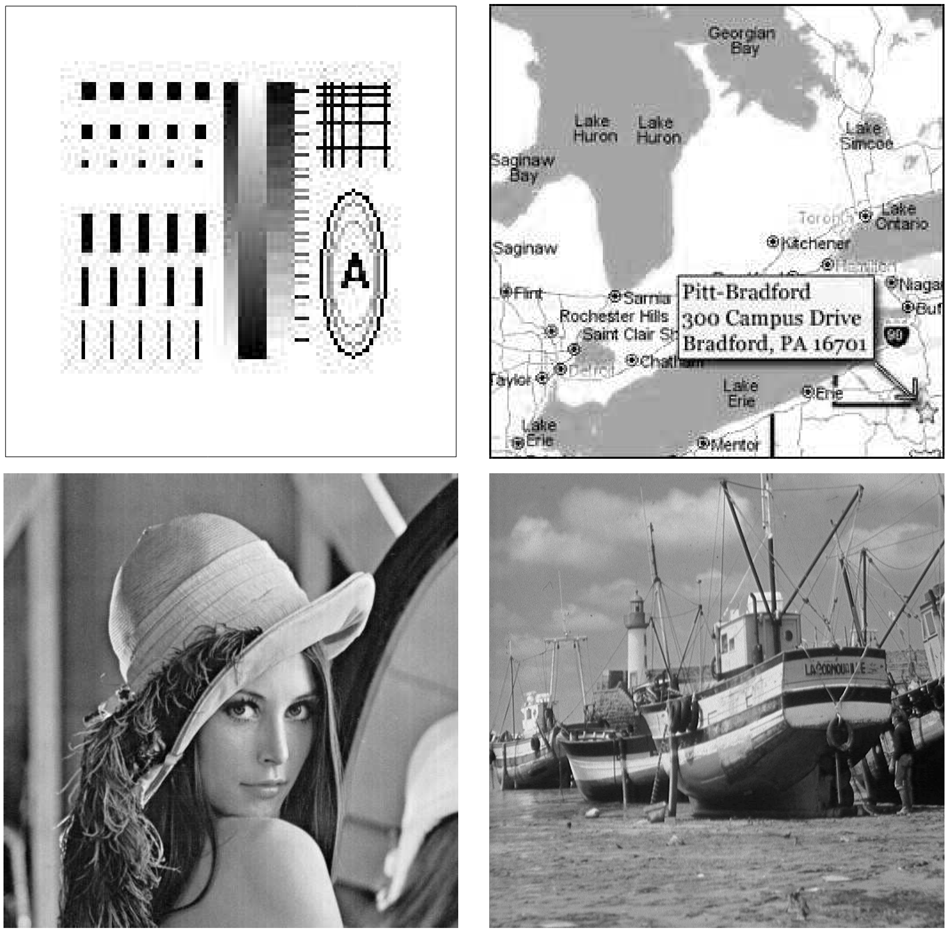

FIGURE 4.1. Original images.

TABLE 4.2. For 512×512 image, different subdomain size with stopping residual $\varepsilon = 10^{-4}$, $\sigma = 10^{-4}$, $\lambda = 0.1$, and $\beta = 10^{-5}$, $\delta = 2$, $S = 5^2$. Each percentage in the "time" column for DD is the ratio of the CPU time of the DD algorithm to that of the algorithm without DD.

image	*type*	d	k	PSNR	Time	*type*	d	k	PSNR	Time
lena512	Gaussian	TV	147	36.5751	1391.7	motion	TV	127	34.5532	1324.1
		8	143	36.4930	15.69%		8	142	34.6034	12.13%
lena512	Gaussian	16	149	36.4378	8.07%	motion	16	128	34.5034	7.23%
		32	146	36.4967	6.62%		32	128	34.4894	5.99%
		64	147	36.5066	9.77%		64	128	34.5777	8.96%
		128	142	36.7895	11.54%		128	127	34.5374	10.85%

5. CONCLUSION.

In this paper, we propose two fast domain decomposition based algorithms for solving the classical total variation based image deblurring model. By partitioning the entire image domain into overlapping subdomains, we are able to solve a sequence of boundary value problems efficiently in

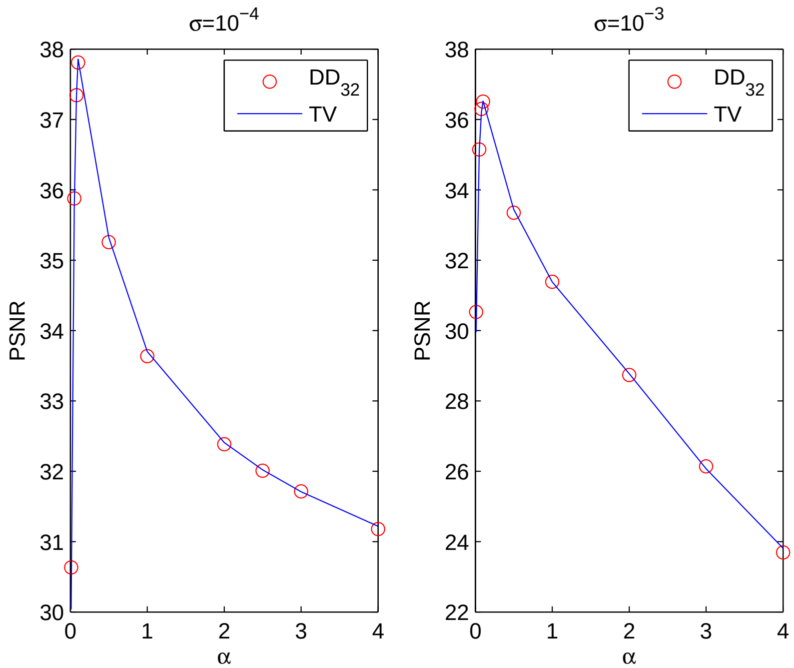

FIGURE 4.2. Choosing a best λ for the different noise level

TABLE 4.3. For 1024×1024 image, different subdomain size with stopping residual $\varepsilon = 10^{-4}$, $\sigma = 10^{-4}$ $\lambda = 0.1$ and $\beta = 10^{-5}$, $\delta = 2$, $S = 5^2$. The elapsed time of DD is showed by the ratio of the time of DD to that of algorithm without DD.

image	*type*	*d*	*k*	PSNR	Time	*type*	*d*	*k*	PSNR	Time
boat1024	Gaussian	TV	131	37.2863	2701.0	motion	TV	116	35.7937	2527.1
boat1024	Gaussian	8	127	37.3214	33.72%	motion	8	111	35.8625	31.68%
		16	129	37.2895	16.88%		16	113	35.8325	15.72%
		32	131	37.2536	14.47%		32	115	35.8342	13.53%
		64	131	37.2718	16.52%		64	115	35.8156	15.38%
		128	131	37.3504	23.13%		128	115	35.7767	21.52%

TABLE 4.4. : TV ALM DD-Deblurring-Different subdomain size with stopping residual $\varepsilon = 10^{-4}$, $\sigma = 10^{-4}$, $\lambda = 2, r = 2$, and $\beta = 0$, and $S = 5^2$.

image	*type*	*d*	*k*	PSNR	Time	*type*	*d*	*k*	PSNR	Time
lena512	Gaussian	8	108	27.3327	289.73	motion	8	134	28.3687	356.48
		16	111	27.2767	152.03		16	138	28.3082	185.21
		32	113	27.1485	131.00		32	142	28.1691	158.78
		64	115	26.8596	144.81		64	142	27.8753	176.96
		128	110	26.8518	214.91		128	142	27.6761	276.06

parallel. More importantly, the proposed algorithms can even be extended to solve certain nonlinear stiff differential equations corresponding to variational image processing models. Various numerical experiments demonstrate that the proposed algorithms perform consistently more efficient than the

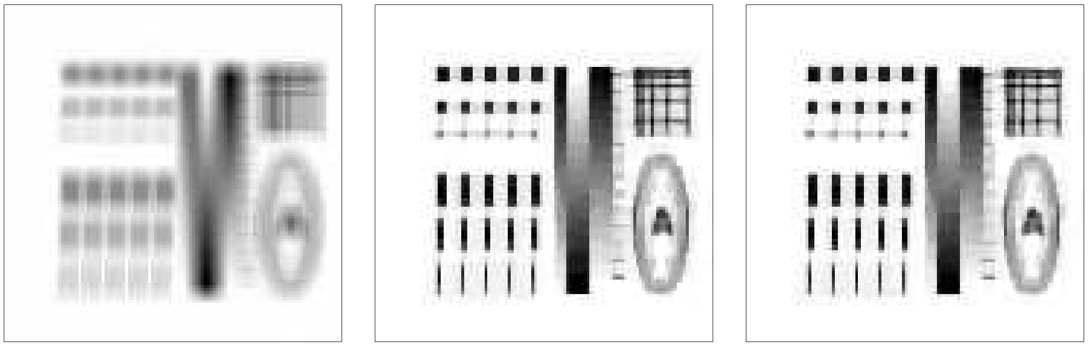

FIGURE 4.3. 128 Gaussian blur: blur (left), restored by TV (middle), restored by DD (right)

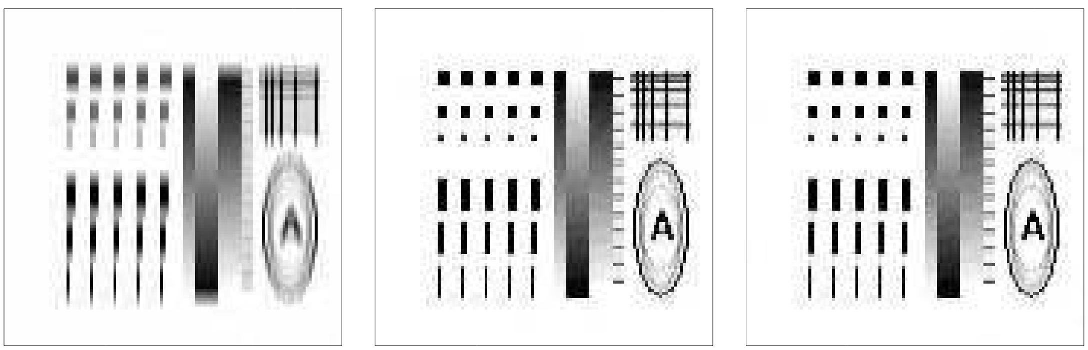

FIGURE 4.4. 128 Motion blur: blur (left), restored by TV (middle), restored by DD (right)

FIGURE 4.5. 256 Motion blur: blur (left), restored by TV (middle), restored by DD (right)

traditional method in terms of restored image quality and CPU time. Furthermore, this work has large potential for a distributed and parallel computation in solving large scale problems in high dimensions.

FIGURE 4.6. Deconvolution results on the Lena image of size 512×512 with Gaussian blur with support size=3: blur (left), restored by DD (right)

FIGURE 4.7. 1024 Gaussian blur with support size = 5: blur (left), restored by DD (right)

TABLE 4.5. **matlabpool**: the parallel efficiency (E) tests for the boat1024 image with different subdomain size with tolerance $\varepsilon = 10^{-4}$, $\sigma = 10^{-4}$, $\lambda = 0.1$ and $\beta = 10^{-5}$, $\delta = 5$.

type	d	k	PSNR	Time	E	*type*	d	k	PSNR	Time	E
Gaussian	8	167	30.4235	1180.52	0.725	motion	8	166	31.4299	1145.45	0.7175
	16	174	30.2730	561.74	0.8		16	167	31.4690	515.74	0.85
	32	173	30.4489	424.37	0.85		32	171	31.4479	466.58	0.79
	64	174	30.4364	482.82	0.85		64	174	31.3900	462.32	0.855
	128	176	30.4092	639.38	0.8		128	173	31.3835	613.06	0.8325
	256	177	30.5370	882.09	0.625		256	172	31.4625	846.09	0.7175

ACKNOWLEDGMENTS.

The first two authors would like to thank MAS, and SPMS for the invitation of visit Nanyang Technological University in Singapore. The first author would also like to thank Prof. X.C. Tai and

L.L. Wang for the helpful discussion and advice. The authors would like to thank the anonymous referees for their valuable comments which helped to improve the manuscript.

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] A.C. Bovik. *Handbook of image and video processing (communications, networking and multimedia)*. Academic Press, Inc. Orlando, FL, USA, 2005.

[3] 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.

[4] C. Brito-Loeza and K. Chen. Multigrid algorithm for high order denoising. *SIAM Journal on Imaging Sciences*, 3(3):363–389, 2010.

[5] C.A.Micchelli, L.X.Shen, and Y.S.Xu. Proximity algorithms for image models: denoising. *Inverse Problems*, 27(4):45009–45038, 2011.

[6] 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.

[7] H. Chang, X. Zhang, Tai X-C, and D. Yang. Domain decomposition methods for nonlocal total variation image restoration. *J.Sci.Comput. to appear*.

[8] Q. S. Chang, W. C. Wang, and J. Xu. A method for the total variation-based reconstruction of noisy and blurred image. Springer, 2007.

[9] 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.

[10] M. Fornasier, A. Langer, and C. B. Schönlieb. Domain decomposition methods for compressed sensing. *arXiv preprint arXiv:0902.0124*, 2009.

[11] M. Fornasier, A. Langer, and C.B. Schönlieb. A convergent overlapping domain decomposition method for total variation minimization. *Numerische Mathematik*, 116(4):645–685, 2010.

[12] M. Fornasier and C. B. Schönlieb. Subspace correction methods for total variation and ℓ_1-minimization. *SIAM Journal on Numerical Analysis*, 47(5):3397–3428, 2009.

[13] M. Griebel and P. Oswald. On the abstract theory of additive and multiplicative Schwarz algorithms. *Numerische Mathematik*, 70(2):163–180, 1995.

[14] 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.

[15] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.

[16] Y.Y. Shi and Q.S. Chang. Efficient algorithm for isotropic and anisotropic total variation deblurring and denoising. *Journal of Applied Mathematics*, 2013, 2013.

[17] 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.

[18] X.C. Tai and Y. Duan. Domain decomposition methods with Graph cuts algotithms for image segmentation. *UCLA CAM Report*, 9–54, 2009.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] C.R. Vogel and M.E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.

[24] C. L. Wu and X. C. Tai. Augmented lagrangian method, dual methods and split-bregman iterations for rof, vectorial tv and higher order models. *SIAM J. Imaging Sci.*, 3(3):300–339, 2010.

[25] J. Xu, X. C. Tai, and L. L. Wang. A two-level domain decomposition method for image restoration. *Inverse problems and Image*, 4(3):523–545, 2010.

[26] J.C. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, 34(4):581–613, 1992.

[27] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1(1):143–168, 2008.