Parallel domain decomposition based algorithm for large scale color image denoising*

Haiwei Fu
School of Mathematics and Statistics
Xi'an Jiaotong University
Xi'an 710049, Shaanxi, P. R. China
fuhaiwei0717@stu.xjtu.edu.cn

Rongliang Chen**
Shenzhen Institutes of Advanced Technology
Chinese Academy of Sciences
Shenzhen 518055, Guangdong, P. R. China
rl.chen@siat.ac.cn

Rongmin Chen
Zhejiang Giian Test Institute Co., Ltd.
Hangzhou 310021, P. R. China
crm@giian.com

***Abstract*— An effective method for recovering blocky and discontinuous images from noisy data is the Total variation (TV) method. In this paper, we introduce a parallel domain decomposition based Newton-Krylov-Schwarz (NKS) method to numerically solve the total variation minimization model for the color image restoration. We show numerically that the NKS method converges and scales well on parallel computers with over one thousand processors. Several numerical experiments and comparisons demonstrate that the proposed method is fast and robust, especially for large scale color images.**

***Index Terms*— Color image restoration, Newton-Krylov-Schwarz, total variation, parallel processing.**

I. Introduction

Based on the total variation (TV) norm introduced in [1], several approaches have been introduced for intensity images denoising [2, 3]. Since TV based algorithms are proved effective for denoising intensity images, it is natural to extend them to color images. The color image denoising problem has attracted lots of attention recently [4–7]. A color image can be represented as a three-component intensity function with red, green and blue. In applications, the observed image is usually distorted by some additive noise such as the Gaussian white noise. In this paper, the original green, red, and blue scenes are denoted as $u_g(x,y)$, $u_r(x,y)$ and $u_b(x,y)$, the corresponding observed scenes are denoted as $z_g(x,y)$, $z_r(x,y)$ and $z_b(x,y)$, and the noise present in the observed image are denoted as $v_g(x,y)$, $v_r(x,y)$ and $v_b(x,y)$, and we also define

$$\begin{cases} \mathbf{z} = (z_r\ z_g\ z_b)^T, \\ \mathbf{u} = (u_r\ u_g\ u_b)^T, \\ \mathbf{v} = (v_r\ v_g\ v_b)^T. \end{cases} \tag{1}$$

Then the image with additive noise is given as:

$$\mathbf{z} = \mathbf{u} + \mathbf{v}. \tag{2}$$

Suppose that $\mathbf{u}$ is defined on a bounded domain Ω. Similar to the TV model of an intensity image, the unconstrained total variation color image restoration problem is described as [8]:

$$\min_{\mathbf{u}\in BV(\Omega)} \left\{ \int_\Omega \alpha|\nabla\mathbf{u}| d\Omega + \frac{1}{2}\|\mathbf{u}-\mathbf{z}\|_2^2 \right\}, \tag{3}$$

where the TV regularization term $|\nabla\mathbf{u}|$ is given by

$$|\nabla\mathbf{u}| = \sqrt{\sum_{k\in r,g,b} |(\nabla u_k)_x|^2 + |(\nabla u_k)_y|^2}, \tag{4}$$

and α is a positive regularization parameter which controls the trade-off between the fidelity of the data and the variability in $\mathbf{u}$. In the actual computation, to avoid the singularity of the Jacobian of the problem, a small positive parameter β is introduced to the TV functional as:

$$\min_{\mathbf{u}\in BV(\Omega)} \left\{ \int_\Omega \alpha\sqrt{|\nabla\mathbf{u}|^2+\beta} d\Omega + \frac{1}{2}\|\mathbf{u}-\mathbf{z}\|_2^2 \right\}. \tag{5}$$

To solve the optimization problem (5), a standard way is to solve the associated Euler-Lagrange equation obtained by the Fre*́*chet derivative of (5) as shown in [8]:

$$u_k - z_k - \alpha\nabla\cdot\left(\frac{\nabla u_k}{\sqrt{|\nabla\mathbf{u}|^2+\beta}}\right) = 0 \qquad in\ \Omega, \tag{6}$$

with a homogeneous Neumann boundary condition:

$$\frac{\partial u_k}{\partial\mathbf{n}} = 0, \tag{7}$$

where $k \in \{r,g,b\}$. Since β is a perturbation parameter, the smaller β is, the higher quality of the restoration of image edges is achieved, however, the convergence is slower because (6) becomes closer to be singular. The solution of the nonlinear equation (6) is challenging because of the ill-conditioness and high nonlinearity, and the high computational cost is a bottleneck, which prevents its widespread use in practice for large color image processing. With the rapid development of the supercomputer, parallel computing provides a way for the large-scale image processing.

The main aim of this paper is to study a solution method for the color image restoration problem based on the total variation minimization method. We propose a parallel domain decomposition based Newton-Krylov-Schwarz (NKS) algorithm [9] for the solution of (6), where an inexact Newton

*The research was supported in part by NSF of China under grant 11271069 and 11226105, the Shenzhen Peacock Plan (No. KQCX20130628112914303). **Corresponding author: Rongliang Chen (rl.chen@siat.ac.cn)

978-1-4799-4808-6 /14/$31.00 ©2014 IEEE

method with an analytical Jacobian matrix is used as the nonlinear solver, in each Newton step, a preconditioned Krylov subspace method is used to solve the Jacobian system, and a restricted additive Schwarz (RAS) preconditioner [10] is used to accelerate the convergence. We mainly focus on the efficiency and parallel scalability of the algorithm.

The outline of this paper is as follows. In Section 2, we present the discretization of the TV-denoising model of color images and we introduce the NKS algorithm in Section 3. In Section 4, some numerical experiments are shown to demonstrate the efficiency and robustness of the algorithm and some conclusions and discussions are followed in Section 5.

II. Discretization

The image domain $\Omega = (0,1) \times (0,1)$ is partitioned into $N \times N$ uniform meshes with the mesh size $h = 1/N$ according to the number of pixels. We denote the grid points as

$$(x_i, y_j) = ((i-1)h, (j-1)h), \qquad 1 \leq i, j \leq N+1, \quad (8)$$

the pixel value of the observed image $\mathbf{z}$ and the imaginary ideal image $\mathbf{u}$ at point (x_i, y_j) as $\mathbf{z}_{i,j}$ and $\mathbf{u}_{i,j}$, respectively. A finite difference method is used to discretize (6). For $k \in \{r, g, b\}$, we define

$$\begin{cases} \delta_x^{\pm}(u_k)_{i,j} = \pm((u_k)_{i\pm1,j} - (u_k)_{i,j}), \\ \delta_y^{\pm}(u_k)_{i,j} = \pm((u_k)_{i,j\pm1} - (u_k)_{i,j}), \\ \delta_x^{c}(u_k)_{i,j} = 0.5((u_k)_{i+1,j} - (u_k)_{i-1,j}), \\ \delta_y^{c}(u_k)_{i,j} = 0.5((u_k)_{i,j+1} - (u_k)_{i,j-1}), \\ |\delta_x^c \mathbf{u}_{i,j}|^2 = \sum\limits_{k\in\{r,g,b\}} (\delta_x^c(u_k)_{i,j})^2, \\ |\delta_y^c \mathbf{u}_{i,j}|^2 = \sum\limits_{k\in\{r,g,b\}} (\delta_y^c(u_k)_{i,j})^2, \\ |\nabla \mathbf{u}_{i,j}|^2 = |\delta_x^c \mathbf{u}_{i,j}|^2 + |\delta_y^c \mathbf{u}_{i,j}\ |^2. \end{cases} \quad (9)$$

Then, for $k \in \{r, g, b\}$, the finite difference approximation of (6) at the interior point (x_i, y_j) reads as

$$\begin{aligned} &(f_k(\mathbf{u}))_{i,j} = (u_k)_{i,j} - (z_k)_{i,j} - \\ &\alpha_h \{ \frac{\delta_x^+(u_k)_{i,j}}{\sqrt{|\nabla \mathbf{u}_{i,j}|^2 + (\delta_x^+(u_k)_{i,j})^2 - (\delta_x^c(u_k)_{i,j})^2 + \beta_h}} \\ &- \frac{\delta_x^-(u_k)_{i,j}}{\sqrt{|\nabla \mathbf{u}_{i,j}|^2 + (\delta_x^-(u_k)_{i,j})^2 - (\delta_x^c(u_k)_{i,j})^2 + \beta_h}} \\ &+ \frac{\delta_y^+(u_k)_{i,j}}{\sqrt{|\nabla \mathbf{u}_{i,j}|^2 + (\delta_y^+(u_k)_{i,j})^2 - (\delta_y^c(u_k)_{i,j})^2 + \beta_h}} \\ &- \frac{\delta_y^-(u_k)_{i,j}}{\sqrt{|\nabla \mathbf{u}_{i,j}|^2 + (\delta_y^-(u_k)_{i,j})^2 - (\delta_y^c(u_k)_{i,j})^2 + \beta_h}} \} = 0, \end{aligned} \quad (10)$$

where $\alpha_h = \alpha/h$ and $\beta_h = \beta h^2$, for simplicity, we drop the subscript in the rest of the paper. The one-sided first-order finite differences are used to treat the Neumann boundary conditions.

After the discretization, with a natural ordering of the unknowns $\mathbf{u}_{i,j}$, the partial differential equation (6) is transferred into an algebraic nonlinear system

$$f(\mathbf{u}) = 0. \quad (11)$$

Next we introduce a NKS algorithm for solving this nonlinear system.

III. Newton-Krylov-Schwarz Algorithm

The NKS algorithm is composed with Newton, Krylov and Schwarz methods, where an inexact Newton method is used as the nonlinear solver, a Krylov subspace method is employed as the linear solver in each Newton step, and a Schwarz type preconditioner is introduced to accelerate the convergence. The NKS algorithm is described as follows.

Algorithm 1 : NKS

```
Use the observed image as the initial guess $\mathbf{u}^0$
For $k = 0, 1, \ldots$ until convergence, do
  • Construct the Jacobian matrix $J(\mathbf{u}^k)$
  • Solve the following right-preconditioned Jacobian system inexactly by a Krylov subspace method
        $J(\mathbf{u}^k) M_k^{-1} M_k \mathbf{s}^k = -f(\mathbf{u}^k)$    (12)
  • Do a cubic line search to find a step length $\tau_k$
  • Set $\mathbf{u}^{k+1} = \mathbf{u}^k + \tau_k \mathbf{s}^k$
End
```

Here $J(\mathbf{u}^k)$ is the Jacobian of $f(\mathbf{u})$ at point $\mathbf{u}^k$, M_k^{-1} is an additive Schwarz preconditioner to be introduced shortly. The inexactness means that the linear system (12) is solved inexactly in the sense of

$$\parallel f(\mathbf{u}^k) + J(\mathbf{u}^k)\mathbf{s}^k \parallel \leq \eta \parallel f(\mathbf{u}^k) \parallel, \quad (13)$$

where $\eta \in [0, 1)$ is a parameter used to control the accuracy of the linear solver. In the NKS algorithm, the most time-consuming step is the solution of the large, sparse and ill-conditioned Jacobian system (12) which is solved by a preconditioned GMRES method [11] in this paper. The preconditioner plays a very important role, without which the GMRES method may converge very slowly or even diverge, and a good choice of preconditioner accelerates the convergence significantly. In this paper, we use an overlapping restricted additive Schwarz (RAS) preconditioner [10, 12].

To define the RAS preconditioner M_k^{-1}, one needs an overlapping partition of the computational domain Ω. We first partition Ω into n_p non-overlapping subdomains $\Omega_l (l = 1, \ldots, n_p)$, as shown in Fig. 1, and then each subdomain Ω_l is extended to an overlapping subdomain $\Omega_l^{'}$ by including δ layers of mesh cells belonging to its neighbors. On the internal subdomain boundary $\partial\Omega_l^{'} \backslash \partial\Omega$, the zero Dirichlet boundary conditions are imposed. The RAS preconditioner is then defined as:

$$M_k^{-1} = \sum_{j=1}^{n_p} (R_j^0)^T B_j^{-1} R_j^{\delta}. \quad (14)$$

Here B_j^{-1} is a preconditioner of the local Jacobian matrix J_j

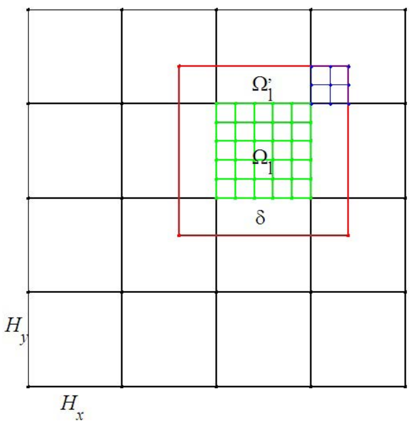

Fig. 1. A diagram and notation for overlapping Schwarz preconditioner. The black solid lines indicate the non-overlapping partition; the red solid rectangle indicates the overlapping partition.

defined on the overlapping subdomain Ω_j'. In this paper, B_j^{-1} is obtained by a LU or an incomplete LU (ILU) factorization of the subdomain Jacobian matrix J_j [11]. R_j^0 and R_j^δ are two restriction operators from the whole domain Ω to the non-overlapping subdomain Ω_j and the overlapping subdomain Ω_j', respectively.

IV. Numerical Results

In this section, some numerical results are presented. We mainly focus on the efficiency, scalability and robustness of the algorithm. Suppose that the pixel values of each channel lie in the interval [0, 1], and the Gaussian white noise is added by the Matlab function *imnoise (I, 'gaussian', M, σ^2)*. In the experiments, the peak signal-to-noise ratio (PSNR) is used as a criterion for the quality of the image restoration which is defined as:

$$PSNR = 10log_{10}\frac{3N^2}{\sum\limits_{k\in\{r,g,b\}}\sum\limits_{i,j}((\hat{u}_k)_{i,j}-(u_k)_{i,j})^2},$$

where $((\hat{u}_k)_{i,j}-(u_k)_{i,j})$ represents the difference of the pixel values between the restored and original clean images, and N is the number of pixels in the coordinate direction. Our software is implemented by using PETSc [13] of Argonne National Laboratory. All computations are carried out on the DAWNING NEBULAE supercomputer at the National Supercomputer Center at Shenzhen, where each compute node has a dual six-core Intel Xeon X5650@2.76GHz CPU and 24GB of memory.

In our numerical tests, in order to get a successful run of the NKS algorithm, a number of parameters need to be specified. Some of them are listed as follows:

1) Nonlinear solver. The initial guess $\mathbf{u}^0$ is the observed noisy image and the relative tolerance $rtol = 10^{-4}$ is used as the stopping condition.
2) Linear solver. We use a restarted GMRES (k) method as the linear solver with the restart parameter $k = 30$. The relative tolerance $\eta = 10^{-4}$ (note that we use zero initial guess for the linear iterative solver) is used for the linear solver at each Newton iteration.
3) Decomposition of Ω. The number of subdomains is set to be equal to the number of processors.

A. *color image denoising*

We test the NKS algorithm on two typical color images: flower-2048 × 2048 and bird-4096 × 4096. Each channel of the original images has the range $[0, 1]$, and the Gaussian white noise with variance $\sigma^2 = 0.04$ is added to the original images. In this paper, we set $\alpha = 0.3$. The parameter β changes from 10^{-3} to 10^{-6}, and we tabulate in Table I the number of Newton iterations, the number of GMRES iterations and the total compute time in seconds. Results show that PSNR does not improve much with the decrease of β, but the computational cost increases significantly. In the rest of this paper, we fix $\beta = 10^{-4}$. The restored images for flower-2048 × 2048 and bird-4096 × 4096 are shown in Fig. 2 and Fig. 4, respectively.

TABLE I
The impact of the parameter β on the flower-2048 × 2048 test case with $\alpha = 0.3$, $\sigma^2 = 0.04$.

β	PSNR	Newton	GMRES	time (s)
10^{-3}	27.01	11	30	22.96
10^{-4}	27.22	28	237	24.45
10^{-5}	27.18	81	595	157.99
10^{-6}	27.12	126	1394	263.05

B. *Parallel performance and scalability studies*

Now let's see the parallel performance of the algorithm. Results are grouped into tables, and we compare different overlapping size (δ), fill-in levels (γ) of the ILU and the number of processors (n_p), then show their impact on the number of Newton iterations (Newton), the number of linear iterations (GMRES) and the total compute time in seconds (time). To illustrate the impact of δ, we fix $\alpha = 0.3$, $\beta = 10^{-4}$, $n_p = 256$, vary δ from 2 to 10, and tabulate in Table II the numbers of iterations and the total compute time. We find that the overlapping size δ does not impact the number of Newton iterations, but the number of linear iterations decreases as δ increases, and the compute time first decreases, then increases as δ increases. This table suggests that there is an optimal choice of the overlapping size if the goal is to minimize the compute time and the best choice of δ is 4 for this test case.

For the ILU factorization, we test different fill-in levels and results are shown in Table III. We see that the number of Newton iterations keeps in constant with the change of fill-in levels γ, but the number of linear iterations decreases

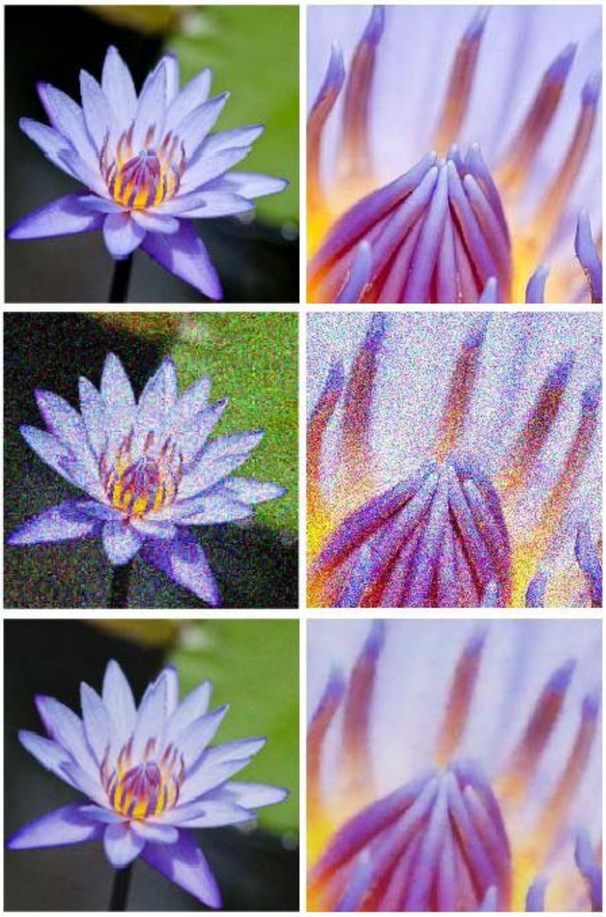

Fig. 2. The flower image denoising results. From top to bottom: the original image, the noisy image with $\delta^2 = 0.04$, and the restored image by NKS algorithm. The whole image is shown on the left and a detailed partial image is on the right. Here $\alpha = 0.3$, $\beta = 10^{-4}$, and $PSNR = 27.2209$.

with the increase of γ, and the compute time first decreases then increases as γ increases.

The parallel performance of our algorithm is shown in Table IV. We solve the subdomain problems by both LU and ILU. For both of the methods, when we increase the number of processors, the number of Newton iterations does not change, the number of GMRES iterations increases reasonably, and the compute time decreases quickly. Fig. 3 shows the speedup of our algorithm for the flower test case. The red line refers to the linear speedup which means that the compute time is halved when the number of processors is doubled, and the other two lines refer to the actual speedup of our algorithm with different local solvers. Table IV and Fig. 3 show that our algorithm has superlinear and nearly linear speedup for LU and ILU methods, respectively.

We also test our algorithm on a larger size color image bird-4096 × 4096, and use the ILU factorization as the local solver, again, the results in Table V show that our method scales well when we increase the number of processors.

TABLE II

THE IMPACT OF THE OVERLAPPING SIZE δ ON THE FLOWER-2048×2048 TEST CASE WITH $\alpha = 0.3$, $\beta = 10^{-4}$, $\sigma^2 = 0.04$.

δ	Newton	GMRES	time (s)
2	28	244	26.83
4	28	237	18.49
6	28	237	20.69
8	28	237	24.64
10	28	237	24.46

TABLE III

THE IMPACT OF THE FILL-IN LEVEL γ OF ILU ON THE FLOWER-2048×2048 TEST CASE WITH $\alpha = 0.3$, $\beta = 10^{-4}$, $\sigma^2 = 0.04$.

γ	Newton	GMRES	time (s)
1	28	302	18.66
2	28	237	18.49
3	28	181	20.03
4	28	151	23.72
5	28	139	24.43

TABLE IV

PARALLEL PERFORMANCE OF NKS ALGORITHM ON FLOWER-2048×2048 TEST CASE WITH $\alpha = 0.3$, $\beta = 10^{-4}$, $\sigma^2 = 0.04$. THE DEGREE OF FREEDOMS (DOF) IS ABOUT 1.2×10^7.

n_p	LU			ILU		
	Newton	GMRES	time(s)	Newton	GMRES	time(s)
16	28	104	1991.61	28	237	215.06
32	28	106	656.26	28	237	109.67
64	28	111	277.01	28	237	57.97
128	28	114	106.45	28	237	32.04
256	28	121	50.65	28	302	18.49

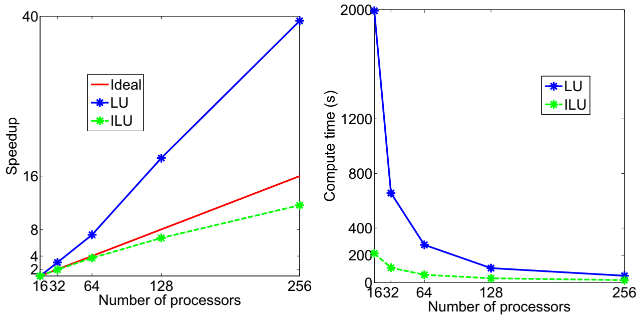

Fig. 3. The speedup and compute time of the Flower-2048×2048 test case with LU and ILU factorization as the local solver. Here $DOF = 1.2 \times 10^7$.

TABLE V

PARALLEL PERFORMANCE OF NKS ALGORITHM ON THE BIRD-4096×4096 TEST CASE WITH $\alpha = 0.3$, $\beta = 10^{-4}$, $\sigma^2 = 0.04$. THE FILL-IN LEVEL γ IS 2 AND THE DOF IS ABOUT 5.0×10^7.

n_p	Newton	GMRES	time (s)
16	33	300	945.08
32	34	311	568.35
64	37	341	310.16
128	37	341	175.41
256	37	340	120.44
512	36	331	55.26
1024	37	341	44.15

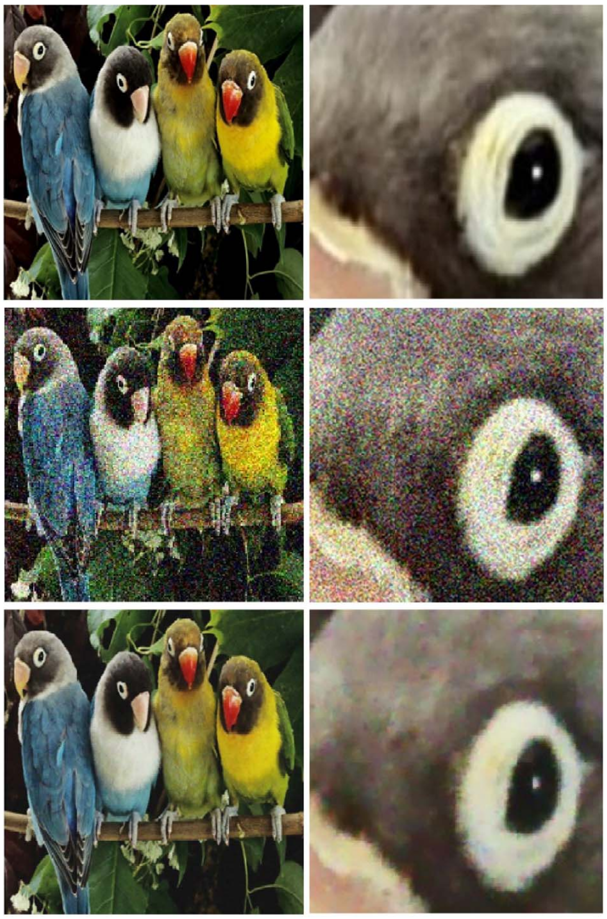

Fig. 4. The bird image denoising results. From top to bottom: the original image, the noisy image with $\delta^2 = 0.04$, and the restored image by NKS algorithm. The whole image is shown on the left and a detailed partial image is on the right. Here $\alpha = 0.3$, $\beta = 10^{-4}$, and PSNR=26.1205.

V. Conclusion

A parallel Newton-Krylov-Schwarz algorithm was developed in this paper for the 2D large scale color image denoising problem based on the TV model. The algorithm is composed with an inexact Newton method as the nonlinear solver, a Krylov subspace method as the linear solver and a restricted additive Schwarz method as a preconditioner. We tested our algorithm on two benchmark problems. Numerical results show that our algorithm can remove the Gaussian white noise efficiently and it has a nearly linear speedup with up to 1024 processors for problems with over 50 million degrees of freedom.

References

[1] L. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithm,” Phys. D., vol. 60, pp. 259-268, 1992.

[2] T. Chan, G. H. Golub, and P. Mulet, “A nonlinear primal-dual method for total variation-based image restoration,” SIAM J. Sci. Comput., vol. 20, pp. 1964-1977, 1999.

[3] C. Vogel and M. Oman, “Iterative methods for total variation denoising,” SIAM J. Sci. Comput., vol. 17, pp. 227-238, 1996.

[4] Y. Wen, M. Ng, and Y. Huang, “Efficient total variation minimization methods for color image restoration,” IEEE Trans. Image Process., vol. 17, pp. 2081-2088, 2008.

[5] J. Yang, W. Yin, Y. Zhang, and Y. Wang, “A fast algorithm for edge-preserving variational multichannel image restoration,” SIAM J. Imaging. Sci., vol. 2, pp. 569-592, 2009.

[6] X. Bresson and T. Chan, “Fast dual minimization of the vectorial total variation norm and applications to color image processing,” Inverse. Probl. Imag., vol. 2, pp. 455-484, 2008.

[7] J. Aujol and S. Kang, “Color image decomposition and restoration,” J. Vis. Commun. Image R., vol. 17, pp. 916-928, 2006.

[8] P. Blomgren and T. Chan, “Color TV: total variation methods for restoration of vector-valued images,” IEEE Trans. Image Process., vol. 7, pp. 304-309, 1998.

[9] S. Ovtchinnikov and X.-C. Cai, “One-level Newton-Krylov-Schwarz algorithm for unsteady nonlinear radiation diffusion problem,” Numer. Lin. Alg. Applics., vol. 11, pp. 867-881, 2004.

[10] X.-C. Cai and M. Sarkis, “A restricted additive Schwarz preconditioner for general sparse linear systems,” SIAM J. Sci. Comput., vol. 21, pp. 792-797, 1999.

[11] Y. Saad, “Iterative Methods for Sparse Linear Systems,” PWS Publishing Company, Boston, 1996.

[12] X.-C. Cai, M. Dryja, and M. Sarkis, “Restricted additive Schwarz preconditioners with harmonic overlap for symmetric positive definite linear systems,” SIAM J. Numer. Anal., vol. 41, pp. 1209-1231, 2003.

[13] S. Balay, et al, “PETSc Users Manual,” Tech. Rep., Argonne National Laboratory, 2012.