

A PARALLEL LINEARIZED ADMM WITH APPLICATION TO MULTICHANNEL TGV-BASED IMAGE RESTORATION

Chuan He*, Changhua Hu

the High-tech Institute of Xi'an,
hechuan8512@163.com

Xuelong Li

Xian Institute of Optics and Precision Mechanics,
Chinese Academy of Sciences

ABSTRACT

A parallel linearized alternating direction method of multipliers (PLADMM) is proposed to solve large-scale imaging inverse problems, which involve the sum of several linear-operator-coupled nonsmooth terms. In the proposed method, the proximity operators of the nonsmooth terms are called individually at each iteration and the auxiliary variables existing in the classical ADMM are excluded. Therefore, the proposed method possesses a highly parallel structure and most of its substeps can be executed simultaneously. The application to multichannel total generalized variation (TGV) based image restoration shows the effectiveness of the proposed method.

Index Terms— Nonsmooth optimization, PLADMM, image restoration, TGV

1. INTRODUCTION

Many inverse problems in imaging, such as image restoration [1–3] and compressed sensing [4], can be formulated as the following large-scale convex optimization problem:

$$\min_{\mathbf{x} \in X} g(\mathbf{x}) + \sum_{h=1}^H f_h(\mathbf{L}_h \mathbf{x}). \quad (1)$$

Denote the set of all convex, proper, and lower semicontinuous functions [5] from Hilbert space X to $\mathbb{R} \cup \{+\infty\}$ as $\Gamma_0\{X\}$. In (1), $g \in \Gamma_0\{X\}$ and $f_h \in \Gamma_0\{V_h\}$ are convex functions whose proximity operators possess closed-forms or at least can be solved efficiently by existent methods. Every $\mathbf{L}_h (X \rightarrow V_h)$ is a bounded linear operator with adjoint \mathbf{L}_h^* and induced norm $\|\mathbf{L}_h\| = \sup \{\|\mathbf{L}_h \mathbf{x}\|_2 : \|\mathbf{x}\|_2 = 1\} < +\infty$. Generally speaking, each function in (1) is either a data fidelity term reflecting the information loss of the observation, or a regularization term imposing some desirable properties on the solution. In pursuit of superior effect, most prevailing regularizers, such as the total variation, are nonsmooth.

The solution of (1) usually suffers from two aspects. First, the data spaces X and V_h in a practical application are typically of high dimension. Second, the function g and the linear-operator-coupled f_h may be nondifferentiable. This often makes the proximal or the operator/variable splitting approach [5] the only viable way to solve (1) [6].

In this paper, a parallel LADMM scheme based on the basic LADMM [7] is proposed to solve the generic problem (1). Compared with the basic LADMM and other prevalent splitting methods [8], the proposed method is typified by the following properties. First, it imposes “full splitting” [6] to (1). At each iteration, only the proximity operators of the convex functions and the linear operators are involved. Therefore, the proposed method possesses a highly parallel structure and can be accelerated by parallel calculation techniques. Second, a specific weight is imposed on every \mathbf{L}_h ; this makes the proposed method more flexible in practical applications. Third, the linear inverse operator, which usually exists in methods dealing with inverse problems [1–4, 9–12], is excluded. Therefore, the proposed method is not partial to a particular data boundary condition. Besides, the proposed method achieves a worst-case $O(1/k)$ convergence rate by exploiting only the first-order information of the functions. In Multichannel image deblurring, where the cross-channel convolution operator cannot be fully diagonalized by FFT within time $O(N \log(N))$, the proposed method holds visible superiority.

In the following, Section II presents the derivation of the PLADMM with a concise convergence analysis. In Section III, the PLADMM is applied to the multichannel TGV-based image restoration and the comparison with several state-of-the-art methods demonstrates its superiority. Finally, some concluding remarks are presented in Section IV.

2. PROPOSED METHOD

The proximity operator of $g \in \Gamma_0\{X\}$ is defined by $\text{prox}_g : \mathbf{x} \rightarrow \arg \min_{\mathbf{x}' \in X} g(\mathbf{x}') + \frac{1}{2} \|\mathbf{x} - \mathbf{x}'\|_2^2$, and its Fenchel conjugate is defined by $g^*(\mathbf{x}) = \sup_{\mathbf{x}' \in X} \{\langle \mathbf{x}, \mathbf{x}' \rangle - g(\mathbf{x}')\}$. We first introduce an auxiliary variable $\mathbf{a}_h \in V_h$ for every $\mathbf{L}_h \mathbf{x}$, i.e., $\mathbf{a}_h = \mathbf{L}_h \mathbf{x}$. Then, similar to the classical ADMM, we get the augmented Lagrangian (AL) functional of (1) as follows:

$$\begin{aligned} \mathcal{L}_{\mathcal{A}}(\mathbf{x}, \mathbf{a}_1, \dots, \mathbf{a}_H; \mathbf{v}_1, \dots, \mathbf{v}_H) = & g(\mathbf{x}) + \\ & \sum_{h=1}^H \left(f_h(\mathbf{a}_h) + \langle \mathbf{v}_h, \mathbf{L}_h \mathbf{x} - \mathbf{a}_h \rangle + \frac{\beta_h}{2} \|\mathbf{L}_h \mathbf{x} - \mathbf{a}_h\|_2^2 \right), \end{aligned} \quad (2)$$

where v_h is the Lagrange multiplier and $\beta_h > 0$ is the penalty parameter. Note that different β_h is employed for each $L_h \mathbf{x}$. The following analysis shows that this operation does not ruin the convergence of the proposed method but makes it more flexible in practice. The proposed PLADMM scheme for finding the saddle point of (2) is as follows:

$$\begin{cases} \mathbf{a}_h^{k+1} = \text{prox}_{f_h/\beta_h} \left(L_h \mathbf{x}^k + \frac{\mathbf{v}_h^k}{\beta_h} \right), h = 1, \dots, H; \\ \mathbf{v}_h^{k+1} = \mathbf{v}_h^k + \beta_h (L_h \mathbf{x}^k - \mathbf{a}_h^{k+1}), h = 1, \dots, H; \\ \mathbf{x}^{k+1} = \text{prox}_{tg} \left(\mathbf{x}^k - t \sum_{h=1}^H \beta_h L_h^* \left(L_h \mathbf{x}^k - \mathbf{a}_h^{k+1} + \frac{\mathbf{v}_h^{k+1}}{\beta_h} \right) \right). \end{cases} \quad (3)$$

In the \mathbf{x} subproblem of (3), we linearize the quadratic term around \mathbf{x}^k [7] by

$$\begin{aligned} \mathbf{x}^{k+1} = & \arg \min_{\mathbf{x}} g(\mathbf{x}) + \\ & \frac{1}{2t} \left\| \mathbf{x} - \mathbf{x}^k + t \sum_{h=1}^H \beta_h L_h^* \left(L_h \mathbf{x}^k - \mathbf{a}_h^{k+1} + \frac{\mathbf{v}_h^{k+1}}{\beta_h} \right) \right\|_2^2. \end{aligned} \quad (4)$$

This operation makes the update of \mathbf{x} becomes a proximity operation of g . For a specific application, if $g = 0$, the prox_{tg} will be removed. Referring to the classical Moreau's decomposition [5], i.e., $\text{prox}_{\beta f^*} \mathbf{v} = \mathbf{v} - \beta \text{prox}_{f/\beta} (\mathbf{v}/\beta)$, we remove the auxiliary variables and get a compacter algorithm to solve (1), which is summarized in Algorithm 1. The convergence analysis will show the parameter selection rule.

Algorithm 1 Parallel Linearized ADMM (PLADMM)

-
- 1: Set $\mathbf{x}^0 = \mathbf{0}$, $\mathbf{v}_h^0 = \mathbf{0}$, $k = 0$, $\beta_h > 0$, $0 < t \leq \frac{1}{\sum_{h=1}^H \beta_h \|L_h^* L_h\|}$;
 - 2: **while** the stopping criterion is not satisfied, **do**
 - 3: $\mathbf{v}_h^{k+1} = \text{prox}_{\beta_h f_h^*} (\beta_h L_h \mathbf{x}^k + \mathbf{v}_h^k)$, $h = 1, \dots, H$;
 - 4: $\mathbf{x}^{k+1} = \text{prox}_{tg} \left(\mathbf{x}^k - t \sum_{h=1}^H L_h^* (2\mathbf{v}_h^{k+1} - \mathbf{v}_h^k) \right)$;
 - 5: $k = k + 1$;
 - 6: **end while** and return \mathbf{x}^{k+1} .
-

With some proper transform, the PLADMM can be recast as a generalization of the basic LADMM, which tackles optimization problems with only two terms. As a consequence, it possesses a worst-case $O(1/k)$ convergence rate.

We denote $\mathbf{a} = (\sqrt{\beta_1} \mathbf{a}_1, \dots, \sqrt{\beta_H} \mathbf{a}_H) \in V \triangleq V_1 \times \dots \times V_H$, $\mathbf{v} = (\mathbf{v}_1/\sqrt{\beta_1}, \dots, \mathbf{v}_H/\sqrt{\beta_H}) \in V$, $L\mathbf{x} = (\sqrt{\beta_1} L_1 \mathbf{x}, \dots, \sqrt{\beta_H} L_H \mathbf{x}) \in V$, and $f(\mathbf{a}) = \sum_{h=1}^H f_h(\mathbf{a}_h) = \sum_{h=1}^H \bar{f}_h(\sqrt{\beta_h} \mathbf{a}_h)$. As a result, $L^* \mathbf{v} = L_1^* \mathbf{v}_1 + \dots + L_H^* \mathbf{v}_H$ holds true. Then the AL (2) can be rewritten as

$$\mathcal{L}'_{\mathcal{A}}(\mathbf{x}, \mathbf{a}; \mathbf{v}) = g(\mathbf{x}) + f(\mathbf{a}) + \langle \mathbf{v}, L\mathbf{x} - \mathbf{a} \rangle + \frac{1}{2} \|L\mathbf{x} - \mathbf{a}\|_2^2. \quad (5)$$

(5) exhibits the form of the AL functional in the classical LADMM. Referring to the convergence analysis of the LADMM [7], we obtain the following convergence theorem for Algorithm 1.

Theorem 1 Let $\mathbf{w} = (\mathbf{x}, \mathbf{a}, \mathbf{v}) \in \Phi \triangleq X \times V \times V$, $\{\mathbf{w}^k\} = \{\mathbf{x}^k, \mathbf{a}^k, \mathbf{v}^k\}$ be the sequence generated by PLADMM with $0 < t \leq \frac{1}{\sum_{h=1}^H \beta_h \|L_h^* L_h\|} \leq \frac{1}{\|L^* L\|}$, and Φ^* be the set of all saddle points of $\mathcal{L}'_{\mathcal{A}}(\mathbf{x}, \mathbf{a}; \mathbf{v})$. Define $\mathbf{P} \triangleq \frac{1}{t} \mathbf{I} - L^* L$ and

$$\|\mathbf{w}\|_G^2 \triangleq \langle \mathbf{x}, \mathbf{P} \mathbf{x} \rangle + \langle \mathbf{a}, \mathbf{a} \rangle + \langle \mathbf{v}, \mathbf{v} \rangle. \quad (6)$$

Then $\{\mathbf{w}^k\}$ converges to a saddle point of $\mathcal{L}'_{\mathcal{A}}(\mathbf{x}, \mathbf{a}; \mathbf{v})$ and the following equation holds:

$$\|\mathbf{w}^k - \mathbf{w}^{k+1}\|_G^2 \leq \frac{1}{k+1} \|\mathbf{w}^0 - \mathbf{w}^*\|_G^2, \quad \forall \mathbf{w}^* \in \Phi^*. \quad (7)$$

In Theorem 1, \mathbf{a}_h^k is implied and can be gained from the second equation in (3). The sufficient condition $t \leq \frac{1}{\|L^* L\|}$ comes from the positive semidefinite property of \mathbf{P} [7]. Due to $L^* L \mathbf{x} = \sum_{h=1}^H \beta_h L_h^* L_h \mathbf{x}$, we have $\|L^* L\| \leq \sum_{h=1}^H \beta_h \|L_h^* L_h\|$ and the restriction of t in PLADMM.

Theorem 1 shows that $\{\mathbf{x}^k, \mathbf{a}_1^k, \dots, \mathbf{a}_H^k, \mathbf{v}_1^k, \dots, \mathbf{v}_H^k\}$ converges to a saddle point of $\mathcal{L}_{\mathcal{A}}(\mathbf{x}, \mathbf{a}_1, \dots, \mathbf{a}_H; \mathbf{v}_1, \dots, \mathbf{v}_H)$, $\{\mathbf{x}^k\}$ converges to a solution of problem (1), and algorithm PLADMM possesses a worst-case $O(1/k)$ convergence rate.

3. MULTICHANNEL TGV-BASED RESTORATION

We consider applying PLADMM to multichannel image deblurring. The regularizer for the ill-posed inverse problem is TGV [13], which is a generalization of the total variation (TV) model by involving high order derivatives of the image function. TGV model is proved to be better than TV model in balancing the edge preservation and the staircasing effect suppression. Although our method can be extended to the T-GV regularizer with an order higher than two, for simplicity, we only treat the second-order TGV and this is sufficient in most cases. Our objective to restore an image is as follows

$$\begin{aligned} (\mathbf{u}^*, \mathbf{p}^*) = & \arg \min_{\mathbf{u}, \mathbf{p}} \alpha_1 \|\nabla \mathbf{u} - \mathbf{p}\|_1 + \alpha_2 \|\mathcal{E} \mathbf{p}\|_1 \quad \text{s.t.} \\ \mathbf{u} \in \Omega \triangleq \{ \mathbf{u} : 0 \leq \mathbf{u} \leq 255 \} \cap \Psi \triangleq \{ \mathbf{u} : \|\mathbf{K} \mathbf{u} - \mathbf{f}\|_2^2 \leq c \}, \end{aligned} \quad (8)$$

where the coupled l_1 terms compose the TGV regularizer $\text{TGV}_{\mathbf{a}}^2$, which degrades to TV with $\alpha_2 = 0$ and $\mathbf{p} = \mathbf{0}$; α_1 and α_2 are fixed positive parameters; \mathbf{f} and \mathbf{u} are the observed and estimated images respectively, both having an $m \times n \times o$ support domain (o -channel) and expressed in vector form; \mathbf{K} is the ill-posed low-pass blur/convolution matrix modeling the acquisition process; Ω and Ψ are the value box constraints (can remarkably improve the quality of images whose pixel values are abundantly lying on the boundary of the given dynamic range [14]) and the data-fidelity constraint (with which we exclude the manual selection of the regularization parameter) respectively. In $\text{TGV}_{\mathbf{a}}^2$, the l_1 penalization of discontinuous components of the image is

allotted to $\alpha_1 \|\nabla \mathbf{u} - \mathbf{p}\|_1$ and the l_1 penalization of regions of moderate slopes is assigned to the other l_1 term. Let $\mathbf{p} \in \mathbb{R}^{mno} \times \mathbb{R}^{mno}$ ($\mathbf{p}_{i,j,l} = (p_{i,j,l,1}, p_{i,j,l,2})$) and \mathcal{E} be the symmetrized difference operator. Then $(\mathcal{E}\mathbf{p})_{i,j,l}$ is given by

$$(\mathcal{E}\mathbf{p})_{i,j,l} = \begin{bmatrix} \nabla_1 p_{i,j,l,1} & \frac{(\nabla_2 p_{i,j,l,1} + \nabla_1 p_{i,j,l,2})}{2} \\ \frac{(\nabla_2 p_{i,j,l,1} + \nabla_1 p_{i,j,l,2})}{2} & \nabla_2 p_{i,j,l,2} \end{bmatrix}, \quad (9)$$

where ∇_1 and ∇_2 denote the difference operators in horizontal and vertical directions. The $\|\cdot\|_1$ s of \mathbf{p} and $\mathcal{E}\mathbf{p}$ are defined as $\|\mathbf{p}\|_1 = \sum_{i,j=1}^{m,n} \sqrt{\sum_{l=1}^o (p_{i,j,l,1}^2 + p_{i,j,l,2}^2)}$ and $\|\mathcal{E}\mathbf{p}\|_1 = \sum_{i,j=1}^{m,n} \sqrt{\sum_{l=1}^o ((\mathcal{E}\mathbf{p})_{i,j,l,1}^2 + (\mathcal{E}\mathbf{p})_{i,j,l,2}^2 + 2(\mathcal{E}\mathbf{p})_{i,j,l,3}^2)}$.

To make use of the proposed PLADMM, we assign $\mathbf{x} = (\mathbf{u}, \mathbf{p})$, $g(\mathbf{x}) = \iota_\Omega(\mathbf{u})$, $f_1(\mathbf{L}_1\mathbf{x}) = \alpha_1 \|\nabla \mathbf{u} - \mathbf{p}\|_1$, $f_2(\mathbf{L}_2\mathbf{x}) = \alpha_2 \|\mathcal{E}\mathbf{p}\|_1$, and $f_3(\mathbf{L}_3\mathbf{x}) = \iota_\Psi(\mathbf{u})$. Define $\tilde{\mathbf{v}}_h^{k+1} \triangleq 2\mathbf{v}_h^{k+1} - \mathbf{v}_h^k$. Then for the solution of (8), we have the following iterative scheme:

$$\left\{ \begin{array}{l} \mathbf{v}_{1,i,j,l}^{k+1} = P_{B_{\alpha_1}} \left(\beta_1 \left((\nabla \mathbf{u}^k)_{i,j,l} - \mathbf{p}_{i,j,l}^k \right) + \mathbf{v}_{1,i,j,l}^k \right); \\ \mathbf{v}_{2,i,j,l}^{k+1} = P_{B_{\alpha_2}} \left(\beta_2 (\mathcal{E}\mathbf{p}^k)_{i,j,l} + \mathbf{v}_{2,i,j,l}^k \right); \\ \mathbf{v}_3^{k+1} = \beta_3 S_{\sqrt{c}} \left(\frac{\mathbf{v}_3^k}{\beta_3} + \mathbf{K}\mathbf{u}^k - \mathbf{f} \right); \\ \mathbf{u}^{k+1} = P_\Omega \left(\mathbf{u}^k - t \left(\nabla^T \tilde{\mathbf{v}}_1^{k+1} + \mathbf{K}^T \tilde{\mathbf{v}}_3^{k+1} \right) \right); \\ \mathbf{p}_1^{k+1} = \mathbf{p}_1^k - t \left(\nabla_1^T \tilde{\mathbf{v}}_{2,1}^{k+1} + \nabla_2^T \tilde{\mathbf{v}}_{2,3}^{k+1} - \tilde{\mathbf{v}}_{1,1}^{k+1} \right); \\ \mathbf{p}_2^{k+1} = \mathbf{p}_2^k - t \left(\nabla_1^T \tilde{\mathbf{v}}_{2,3}^{k+1} + \nabla_2^T \tilde{\mathbf{v}}_{2,1}^{k+1} - \tilde{\mathbf{v}}_{1,2}^{k+1} \right). \end{array} \right. \quad (10)$$

In (10), $P_{B_{\alpha_1}}$ and $P_{B_{\alpha_2}}$ are the two and four dimensional projectors respectively, whereas $S_{\sqrt{c}}$ is an mno dimensional shrinkage operator. The projector P_{B_r} and the shrinkage operator S_r are defined by $P_{B_r}(\mathbf{q}_{i,j,l}) = \min(\|\mathbf{q}_{i,j,l}\|_2, r) \frac{\mathbf{q}_{i,j,l}}{\|\mathbf{q}_{i,j,l}\|_2}$ and $S_r(\mathbf{z}) = \max(\|\mathbf{z}\|_2 - r, 0) \frac{\mathbf{z}}{\|\mathbf{z}\|_2}$, respectively. We denote \mathbf{p}_1 and \mathbf{p}_2 as the combinations of all $p_{i,j,l,1}$ and $p_{i,j,l,2}$, $1 \leq i \leq m, 1 \leq j \leq n, 1 \leq l \leq o$, respectively. Similarly, we define $\mathbf{v}_{1,1}$ and $\mathbf{v}_{1,2}$. Variable \mathbf{v}_2 has the same structure as $\mathcal{E}\mathbf{p}$, and we denote $\mathbf{v}_{2,1}$, $\mathbf{v}_{2,2}$, and $\mathbf{v}_{2,3}$ as the combinations of all $v_{2,i,j,l,1}$, $v_{2,i,j,l,2}$, and $v_{2,i,j,l,3}$, respectively.

The iterative scheme (10) involves no inverse operator. This makes it more attractive in multichannel image processing, where the convolution matrix may not be wholly diagonalized with FFT, due to the existence of cross-channel blurs. In fact, cross-channel blurs prevent most methods handling the image restoration with data-fidelity constraint from extending to multichannel imaging. Besides, (10) possesses a highly parallel structure. The subproblems with respect to \mathbf{v}_1 , \mathbf{v}_2 , and \mathbf{v}_3 are independent to each other and the subproblems with respect to \mathbf{u} , \mathbf{p}_1 , and \mathbf{p}_2 admits the same relationship. Besides, the updates of \mathbf{v}_1 and \mathbf{v}_2 are pixel-wise.

Following the above operator assignment, it holds that $\sum_{h=1}^3 \beta_h \|\mathbf{L}_h^* \mathbf{L}_h\| = \beta_1 \|\nabla^T \nabla\| + \beta_2 \|\mathcal{E}^T \mathcal{E}\| + \beta_3 \|\mathbf{K}^T \mathbf{K}\| \leq 8\beta_1 + 8\beta_2 + \beta_3$. From the basic assumption $\mathbf{K}\mathbf{1} = \mathbf{1}$ [15], we have $\|\mathbf{K}^T \mathbf{K}\| = 1$, where $\mathbf{1}$ is the vector of all 1s. The



Fig. 1. Lena (256 × 256), Peppers (512 × 512), and Monarch (768 × 512).

Table 1. Details of the image deblurring experiment

Problem	Image	Blur kernels	σ	PSNR	SSIM
1	Lena	Set 1	3	20.05	0.5239
2	Peppers	Set 2	8	17.55	0.5140
3	Monarch	Set 3	10	17.95	0.4608

norm analysis of ∇ and \mathcal{E} can be found in [13]. Therefore, we can set $t = \frac{1}{8\beta_1 + 8\beta_2 + \beta_3}$ for the TGV-based image restoration. Besides, we set $(\alpha_1, \alpha_2) = (1, 6)$ for the TGV model.

The upper bound c in (8) is noise-dependent, and we use $c = \tau mn\sigma^2$ to estimate it, where σ^2 is the noise variance and $\tau = -0.0009\text{BSNR} + 0.99$ (obtained by fitting a large number of experimental data on natural images with a straight line) [11]. The blurred signal-to-noise ratio (BSNR) is defined by $\text{BSNR} = 10 \log_{10} (\text{var}(\mathbf{f})/\sigma^2)$ (dB), where $\text{var}(\mathbf{f})$ is the variance of \mathbf{f} . Simply setting $\beta_1 = \beta_2 = \beta_3$ is sufficient for the convergence of the proposed method. However, we experientially set $\beta_1 = \beta_2 = 0.04$ and $\beta_3 = \text{BSNR} \times \beta_1$ to accelerate the convergence. Some more careful tuning of the parameters may leads to better results for a particular application. We compare the TGV-regularized PLADMM (PLADMM-TGV) with the other two state-of-the-art methods in multichannel image deblurring: adaptive parameter estimation for ADMM with TGV regularizer (APEADMM-TGV) [11] and FTVD-v4 [12]. The former one can solve problem (8) without box constraints, whereas the latter one can only treat the unconstrained TV-regularized deconvolution problem. Differing from PLADMM-TGV, they all involve inverse operators like $(\sum \mathbf{L}_h^* \mathbf{L}_h)^{-1}$. Besides, FTVD-v4 involves the manual selection of the regularization parameter which balances the regularization and data-fidelity terms.

Table 2. Comparison in PSNR, SSIM, and CPU time(s).

Problem	Method	PSNR	SSIM	CPU
1	PLADMM-TGV	26.21	0.7680	32.24
	APEADMM-TGV	26.21	0.7649	35.62
	FTVD-v4	26.04	0.7583	10.37
2	PLADMM-TGV	25.57	0.7632	155.46
	APEADMM-TGV	25.56	0.7623	199.93
	FTVD-v4	25.25	0.7507	54.69
3	PLADMM-TGV	23.85	0.8083	234.45
	APEADMM-TGV	23.83	0.8063	302.68
	FTVD-v4	23.61	0.7965	84.43

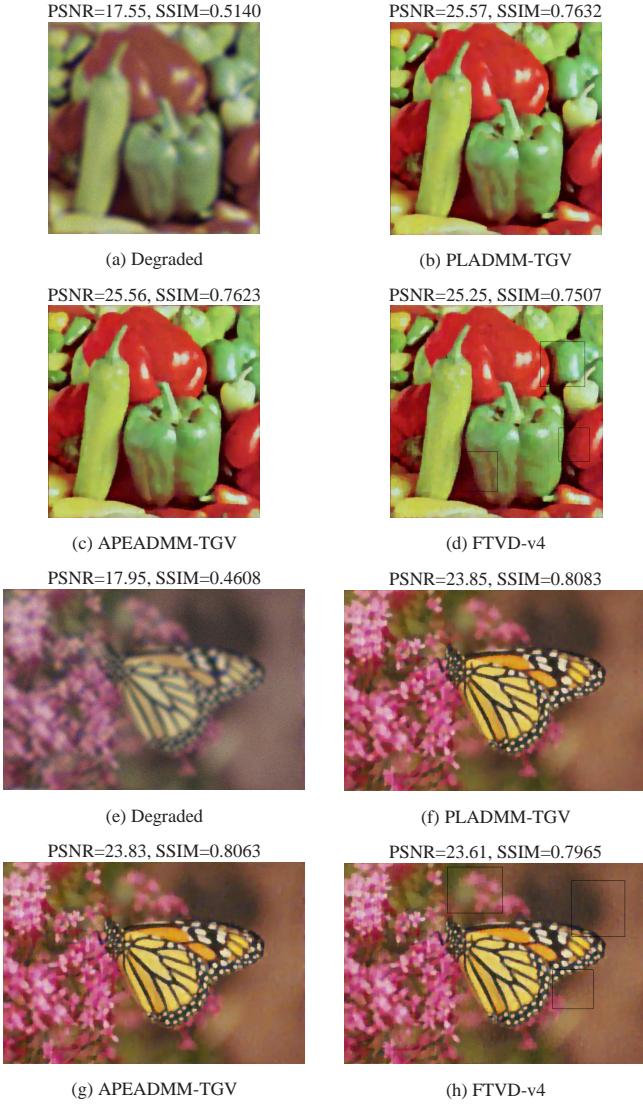


Fig. 2. Degraded and restored Peppers images ((a)–(d)) and degraded and restored Monarch images ((e)–(h)).

The following experiment was performed in MATLAB on a PC with Intel Core (TM) i5 CPU (3.20GHz) and 8GB of RAM. The PSNR (in dB) and the SSIM index [16] are used to estimate the image quality. Since all the three methods admit the $O(1/k)$ convergence rate, we use iteration counter reaching 300, with which all the three methods have sufficiently converged, as the uniform stopping criterion. For simplicity, we denote the average blur kernel with a square blurring size s by $A(s)$, the Gaussian blur kernel with a square blurring size s and a standard deviation δ by $G(s, \delta)$, and the motion blur kernel with a length d and an angle θ by $M(d, \theta)$. We design three experiments elaborated in Table 1 to test the three algorithms. Images Lena (256×256), Peppers (512×512), and Monarch (768×512) are shown in Fig. 1. The three sets

of blurs are generated in the following manner [12]: (1). Generate 9 kernels: $\{A(13), A(15), A(17), G(11, 9), G(21, 11), G(31, 13), M(21, 45), M(41, 90), M(61, 135)\}$; (2). assign the above 9 kernels to $\{K_{11}, K_{12}, K_{13}; K_{21}, K_{22}, K_{23}; K_{31}, K_{32}, K_{33}\}$, where K_{ii} , $i = 1, 2, 3$, are within-channel kernels and the rests are cross-channel kernels; (3). then with the above kernels, we generate the final three sets of blurs for comparison by multiplying relative weights $\{1, 0, 0; 0, 1, 0; 0, 0, 1\}$ (Set 1), $\{0.6, 0.2, 0.2; 0.15, 0.7, 0.15; 0.1, 0.1, 0.8\}$ (Set 2), and $\{0.7, 0.15, 0.15; 0.1, 0.8, 0.1; 0.2, 0.2, 0.6\}$ (Set 3) to the corresponding blur kernels. After blurring the three images, we add Gaussian noise with standard variances shown in Table 1 to each to obtain the observations.

Table 2 exhibits the results in terms of PSNR, SSIM, and CPU time. We observe from Table 2 that, first, PLADMM-TGV and APEADMM-TGV achieve similar PSNRs and SSIMs which are higher than those of FTVD-v4, due to the more sophisticated TGV regularizer. In spite of consuming the least CPU time for one implementation, FTVD-v4 actually costs more time than the other two algorithms, owing to the intractable manual selection of the regularization parameter by try-and-error. Second, compared with APEADMM-TGV, PLADMM-TGV costs less CPU time for problems 2 and 3, where cross-channel blurs are involved and the blur matrix cannot be wholly diagonalized by FFT. This matches our expectation for PLADMM, though its time superiority over APEADMM-TGV is not obvious for problem 1, where only within-channel blurs are involved. On a NVIDIA Tesla K20C GPU, PLADMM-TGV takes only 0.39s, 1.44s, and 1.82s to perform 300 iterations for the three background problems respectively, i.e., the speedup ratios of GPU over CPU for PLADMM-TGV reach the level of $100\times$. This includes all data transfers to the GPU and back.

Fig. 2 presents the degraded and restored images for problem 2 and problem 3 respectively. Fig. 2 shows that PLADMM-TGV and APEADMM-TGV obtain restorations with higher visual quality, by effectively suppressing the staircasing effects existing in the restorations of FTVD-v4 (some of which are labeled with the black rectangles). Since we employ images of different sizes, noises of different variances, and blurs of different types, the robustness of the proposed method is illustrated to a certain extent.

4. CONCLUDING REMARKS

A parallel linearized ADMM is proposed to solve large-scale inverse problems. By further exploiting the structure of the problem with proximal splitting technique, the proposed method admits a compact straightforward frame, which excludes inner loop and inverse linear operator. Due to the parallel structure, the proposed method possesses high potential in parallel calculations. Experimental results validate the efficiency of the proposed method for image restoration.

5. REFERENCES

- [1] X. Zhang, W. Lin, R. Xiong, X. Liu, S. Ma, and W. Gao, “Low-rank decompositon-based restoration of compressed image via adaptive noise estimation,” *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4158–4171, 2016.
- [2] Y. Hu, X. Lu, and M. Jacob, “Multiple degree total variation (mdtv) regularization for image restoration,” in *IEEE International Conference on Image Processing (ICIP)*, Phoenix, USA, 2016, pp. 1958–1962.
- [3] Z. Lu, H. Li, and W. Li, “A bayesian adaptive weighted total generalized variation model for image restoration,” in *IEEE International Conference on Image Processing (ICIP)*, Quebec, Canada, 2015, pp. 492–496.
- [4] N. Eslahi and A. Aghagolzadeh, “Compressive sensing image restoration using adaptive curvelet thresholding and nonlocal sparse regularization,” *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3126–3140, 2016.
- [5] H. Bauschke and P. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, NY, 1rd edition, 2011.
- [6] L. Condat, “A generic proximal algorithm for convex optimization—application to total variation minimization,” *IEEE Signal Proc. Letters*, vol. 21, no. 8, pp. 985–989, 2014.
- [7] B. He and X. Yuan, “On non-ergodic convergence rate of douglas-rachford alternating direction method of multipliers,” *Numerische Mathematik*, vol. 130, no. 3, pp. 567–577, 2015.
- [8] P. Combettes and J. Pesquet, “Proximal splitting methods in signal processing,” in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, and H. Wolkowicz, Eds., pp. 185–212. Springer, New York, NY, 2010.
- [9] C. He, C. Hu, X. Li, X. Yang, and W. Zhang, “A parallel alternating direction method with application to compound l1-regularized imaging inverse problems,” *Inform. Sci.*, vol. 348, pp. 179–197, 2016.
- [10] C. He, C. Hu, X. Li, and W. Zhang, “A parallel primal-dual splitting method for image restoration,” *Inform. Sci.*, vol. 358-359, pp. 73–91, 2016.
- [11] C. He, C. Hu, W. Zhang, and B. Shi, “A fast adaptive parameter estimation for total variation image restoration,” *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 4954–4967, 2014.
- [12] J. Yang, W. Yin, Y. Zhang, and Y. Wang, “A fast algorithm for edge-preserving variational multichannel image restoration,” *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 569–592, 2009.
- [13] K. Bredies, K. Kunisch, J. Liu, and T. Pock, “Total generalized variation,” *SIAM J. Imag. Sci.*, vol. 3, no. 3, pp. 492–526, 2010.
- [14] R. Chan and J. Ma, “A multiplicative iterative algorithm for box-constrained penalized likelihood image restoration,” *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3168–3181, 2012.
- [15] T. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, SIAM, Philadelphia, 1rd edition, 2005.
- [16] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image aquality assessment: from error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.