

Fast Algorithms for the Anisotropic LLT Model in Image Denoising

Zhi-Feng Pang^{1,2}, Li-Lian Wang^{*,2} and Yu-Fei Yang³

¹ College of Mathematics and Information Science, Henan University, Kaifeng, 475004, China.

² Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, 637371, Singapore.

³ College of Mathematics and Econometrics, Hunan University, Changsha, 410082, China.

Received 23 December 2010; Accepted (in revised version) 26 April 2011

Available online 27 July 2011

Abstract. In this paper, we propose a new projection method for solving a general minimization problems with two L^1 -regularization terms for image denoising. It is related to the split Bregman method, but it avoids solving PDEs in the iteration. We employ the fast iterative shrinkage-thresholding algorithm (FISTA) to speed up the proposed method to a convergence rate $O(k^{-2})$. We also show the convergence of the algorithms. Finally, we apply the methods to the anisotropic Lysaker, Lundervold and Tai (LLT) model and demonstrate their efficiency.

AMS subject classifications: 65M10, 78A48

Key words: Image denoising, anisotropic LLT model, Douglas-Rachford splitting method, split Bregman method, projection method, fast projection method.

1. Introduction

Image denoising is a fundamental task in image processing, which aims to recover a noise-free image u from a noise polluted image f . In general, it can be modeled by $f = u + \eta$, where η is the unknown noise component. Among various methods for finding such a decomposition, the variational approach is to restore u by solving the minimization problem (see, e.g., [2, 14]):

$$\min_{u \in X} \{ \mathcal{R}(u) + \lambda \mathcal{F}(u - f) \}, \quad (1.1)$$

where the functionals $\mathcal{F}(\cdot)$ and $\mathcal{R}(\cdot)$ are respectively the data fidelity and regularization terms defined on a suitable functional space X , and $\lambda > 0$ is a parameter to balance two

*Corresponding author. Email addresses: zhifengpang@163.com (Z.-F. Pang), lilian@ntu.edu.sg (L.-L. Wang), yfyang@hnu.edu.cn (Y.-F. Yang)

terms. The popular total variation (TV) regularized L^2 -model, known as the Rudin-Osher-Fatemi (ROF) model [35], takes the form

$$\min_{u \in BV(\Omega)} \left\{ \int_{\Omega} |Du| + \frac{\lambda}{2} \int_{\Omega} |u - f|^2 d\Omega \right\}, \quad (1.2)$$

where Ω is a bounded domain in R^2 with the Lipschitz boundary, $BV(\Omega)$ is the space of functions with bounded variation, and $\int_{\Omega} |Du|$ is the total variation of u (see, e.g., [1]). An important variant of the ROF model is as follows (see, e.g., [22, 24, 26]):

$$\min_u \left\{ \int_{\Omega} (|u_x| + |u_y|) d\Omega + \frac{\lambda}{2} \int_{\Omega} |u - f|^2 d\Omega \right\}. \quad (1.3)$$

The models (1.2) and (1.3) are known as the isotropic and anisotropic ROF model, respectively. They function well for noise removal, and simultaneously preserve discontinuities and edges, so they have been extensively used for a variety of image restoration problems (see, e.g., [14, 33]).

However, as pointed out in [5, 11], the ROF model induces the so-called "staircase effect", as its cost functional is borderline convex (with a linear growth with respect to the image gradients), and it oftentimes produces piecewise constant artificial architectures. Some models using higher-order derivative information have been proposed to overcome this drawback (see, e.g., [13, 27, 37, 46]). For instance, Lysaker, Lundervold and Tai [27] suggested the model (termed as the anisotropic LLT model):

$$\min_u \left\{ F(u) := \int_{\Omega} (|u_{xx}| + |u_{yy}|) d\Omega + \frac{\beta}{2} \int_{\Omega} |u - f|^2 d\Omega \right\}, \quad \beta > 0. \quad (1.4)$$

The use of second-order derivatives damps oscillations faster than the total variation regularized model, so (1.4) can reduce the "staircase effect", and produce better approximation to the natural image [27, 46].

Over the past decade, many methods have been developed for the ROF model (1.1). These algorithms typically include (i) the primal approaches, such as artificial time marching algorithms [26, 27, 35], fixed point iterative algorithm [41], and the multigrid method [16]; (ii) the dual methods [9, 10, 12, 15, 30, 32, 38], and (iii) the primal-dual approaches [6, 20, 42], the augmented Lagrangian method [40, 44], and the split Bregman type methods [8, 21, 29, 34, 39, 45]. Moreover, fast graph-cut algorithms [9, 18] have been developed for (1.3).

However, to the best of our knowledge, there are very few discussions on efficient minimization of the anisotropic LLT model (1.4). In this paper, we shall put this model in a more general setting and develop fast algorithms for the minimization problem:

$$\min_u \left\{ E(u) := \int_{\Omega} (|\Lambda_1 u| + |\Lambda_2 u|) d\Omega + \frac{\gamma}{2} \int_{\Omega} |u - f|^2 d\Omega \right\}, \quad (1.5)$$

where Λ_1 and Λ_2 are two bounded linear operators over the admissible function space, and γ is a positive constant. Motivated by the split Bregman method for the ROF model [22],

we shall propose a new projection scheme and the resultant algorithm enjoys the advantages: (i) it inherits the strengths of the split Bregman method; (ii) it avoids solving PDEs, so it saves computational time and memory; (iii) it possesses a convergence rate $O(k^{-2})$, where k is the number of iterations. We shall also provide ample numerical experiments to demonstrate the effectiveness of the proposed algorithm.

The rest of the paper is organized as follows. In Section 2, we propose the iterative algorithms for the general model (1.5), and analyze its convergence. In Section 3, we implement the algorithms for the anisotropic LLT model (1.4). The final section is devoted to the numerical experiments and concluding remarks.

2. Formulation of the Algorithms

In this section, we describe the iterative algorithms for the general model (1.5). Our starting point is to reformulate (1.5) as

$$\begin{aligned} & \min_{d_1, d_2, u} \left\{ |d_1|_{L^1(\Omega)} + |d_2|_{L^1(\Omega)} + \frac{\gamma}{2} \|u - f\|_{L^2(\Omega)}^2 \right\}, \\ & \text{subject to } d_1 = \Lambda_1 u \text{ and } d_2 = \Lambda_2 u, \end{aligned} \tag{2.1}$$

where we used $|\cdot|_{L^1(\Omega)}$ and $\|\cdot\|_{L^2(\Omega)}$ to denote the norm of space $L^p(\Omega)$ with $p = 1, 2$, respectively. Indeed, as with the algorithms for ROF model (cf. [22, 39, 40, 43]), it is essential to decompose the original problem into some subproblems that are easier to be solved. Typical techniques for dealing with the constraint formulation include the split Bregman iteration [22], the augmented Lagrangian method [40], and the Douglas-Rachford splitting method [39], etc.

In what follows, we approximate (2.1) by using the penalty method:

$$\min_{u, d_1, d_2} \left\{ |d_1|_{L^1(\Omega)} + |d_2|_{L^1(\Omega)} + \frac{\gamma}{2} \|u - f\|_{L^2(\Omega)}^2 + \frac{\tau_1}{2} \|d_1 - \Lambda_1 u\|_{L^2(\Omega)}^2 + \frac{\tau_2}{2} \|d_2 - \Lambda_2 u\|_{L^2(\Omega)}^2 \right\}, \tag{2.2}$$

where $\tau_1, \tau_2 > 0$ are two penalty constants.

2.1. Algorithm based on split Bregman iteration

Using the notion of the Bregman iteration (see, e.g., [22]), we resolve the problem (2.2) by solving the following three subproblems (as summarized in Algorithm 2.1):

$$\left\{ \begin{aligned} (u^{k+1}, d_1^{k+1}, d_2^{k+1}) & := \arg \min_{u, d_1, d_2} \left\{ |d_1|_{L^1(\Omega)} + |d_2|_{L^1(\Omega)} + \frac{\gamma}{2} \|u - f\|_{L^2(\Omega)}^2 \right. \\ & \quad \left. + \frac{\tau_1}{2} \|d_1 - \Lambda_1 u - b_1^k\|_{L^2(\Omega)}^2 + \frac{\tau_2}{2} \|d_2 - \Lambda_2 u - b_2^k\|_{L^2(\Omega)}^2 \right\}, \\ b_1^{k+1} & := b_1^k + \Lambda_1 u^{k+1} - d_1^{k+1}, \\ b_2^{k+1} & := b_2^k + \Lambda_2 u^{k+1} - d_2^{k+1}. \end{aligned} \right. \tag{2.3}$$

Algorithm 2.1. The split Bregman method (SBM) for solving (2.1).

- (I) Set $u^0 = f$, $d_1^0 = 0$, $d_2^0 = 0$, $b_1^0 = 0$, $b_2^0 = 0$ and $k = 0$;
 (II) Compute $(u^{k+1}, d_1^{k+1}, d_2^{k+1}, b_1^{k+1}, b_2^{k+1})$ by

$$\begin{aligned} u^{k+1} &:= \arg \min_u \left\{ \frac{\gamma}{2} \|u - f\|_{L^2(\Omega)}^2 + \frac{\tau_1}{2} \|d_1 - \Lambda_1 u - b_1^k\|_{L^2(\Omega)}^2 + \frac{\tau_2}{2} \|d_2 - \Lambda_2 u - b_2^k\|_{L^2(\Omega)}^2 \right\}, \\ d_1^{k+1} &:= \arg \min_{d_1} \left\{ |d_1|_{L^1(\Omega)} + \frac{\tau_1}{2} \|d_1 - \Lambda_1 u^{k+1} - b_1^k\|_{L^2(\Omega)}^2 \right\}, \\ d_2^{k+1} &:= \arg \min_{d_2} \left\{ |d_2|_{L^1(\Omega)} + \frac{\tau_2}{2} \|d_2 - \Lambda_2 u^{k+1} - b_2^k\|_{L^2(\Omega)}^2 \right\}, \\ b_1^{k+1} &:= b_1^k + \Lambda_1 u^{k+1} - d_1^{k+1}, \\ b_2^{k+1} &:= b_2^k + \Lambda_2 u^{k+1} - d_2^{k+1}; \end{aligned}$$

- (III) If the stop criterion is not satisfied, set $k := k + 1$ and go to Step II.

It is clear that the first subproblem in Step II of Algorithm 2.1 can be solved easily from the optimality condition. The solutions of the second and third subproblems can be obtained by

$$d_1^{k+1} = S_{\tau_1}(\Lambda_1 u^{k+1} + b_1^k), \quad d_2^{k+1} = S_{\tau_2}(\Lambda_2 u^{k+1} + b_2^k), \quad (2.4)$$

where S_τ is the shrinkage operator defined by

$$S_\tau(c) := \frac{c}{|c|_{L^1(\Omega)}} \max\{|c|_{L^1(\Omega)} - \tau^{-1}, 0\}. \quad (2.5)$$

Here, we adopt the convention: $0 \cdot (0/0) = 0$ (cf. [42]).

In fact, the above split Bregman-based method can be regarded as a special case of the Douglas-Rachford splitting algorithm [19], as observed by Setzer and Teuber [39]. To analyze Algorithm 2.1, we recall the convergence of the (generalized) Douglas-Rachford splitting algorithm given by Combettes [17].

Lemma 2.1. *Let X be a Hilbert space, and let $A, B : X \rightarrow 2^X$ be two maximal monotone operators. Define $J_{\mu A} = (I + A/\mu)^{-1}$, for any $\mu > 0$. Assume that a solution v^* of*

$$0 \in A(v) + B(v)$$

exists. Then for any initial guesses t^0, v^0 and $\mu > 0$, the following Douglas-Rachford splitting iteration

$$\begin{cases} t^{k+1} := J_{\mu A}(2v^k - t^k) + t^k - v^k, \\ v^{k+1} := J_{\mu B}(t^{k+1}), \quad k \geq 0, \end{cases} \quad (2.6)$$

converges weakly to an element t^ . Moreover, there holds $v^* := J_{\mu B}(t^*)$ satisfying $0 \in A(v^*) + B(v^*)$. In particular, if X is finite dimensional, the sequence $\{v^k\}$ converges to v^* .*

With the aid of Lemma 2.1, we are able to show the convergence of Algorithm 2.1, as stated in Theorems 2.1 and 2.2, whose proofs are given in Appendix A and Appendix B, respectively.

Theorem 2.1. *The sequence $\{(u^{k+1}, d_1^{k+1}, d_2^{k+1}, b_1^{k+1}, b_2^{k+1})\}$ generated by Algorithm 2.1 is convergent.*

Theorem 2.2. *Assume that $(u^*, d_1^*, d_2^*, b_1^*, b_2^*)$ is the limit point of the sequence $\{(u^k, d_1^k, d_2^k, b_1^k, b_2^k)\}$ generated by Algorithm 2.1. Then u^* is the solution of the constrained optimization problem (1.5).*

The following properties are a direct consequence of Theorem 2.2.

Corollary 2.1. *Assume that u^* is the solution of (1.5) and $\{u^k\}$ is generated by Algorithm 2.1. Then the following properties for Algorithm 2.1 hold:*

$$\lim_{k \rightarrow +\infty} \|u^k - u^*\|_{L^2(\Omega)} = 0, \quad \lim_{k \rightarrow +\infty} \Lambda_1 u^k = \Lambda_1 u^* \quad \text{and} \quad \lim_{k \rightarrow +\infty} \Lambda_2 u^k = \Lambda_2 u^*.$$

2.2. A new projection scheme

The split Bregman method has been widely applied to signal processing and image restoration problems [7, 8, 21–23, 28, 31], but it usually requires to solve PDEs in the inner loop that increases computational time and memory requirement. In this subsection, we propose a projection method based on the relationship between the projection operator and the shrinkage operator. This leads to efficient algorithms that enjoy the advantages of the split Bregman method without solving PDEs.

Definition 2.1. Let X, Y be two Hilbert spaces. The projection operator $P_{B_\tau}(\cdot) : X \rightarrow Y$ onto the closed disc $B_\tau := \{c \in X : \|c\|_{L^1(\Omega)} \leq \tau^{-1}\}$ is defined by

$$P_{B_\tau}(c) = \frac{c}{\|c\|_{L^1(\Omega)}} \min(\|c\|_{L^1(\Omega)}, \tau^{-1}), \tag{2.7}$$

where $\tau > 0$.

In view of the definitions (2.5) and (2.7), we have the following identity:

$$P_{B_\tau}(c) + S_\tau(c) = c, \quad \forall c \in X, \quad \forall \tau > 0. \tag{2.8}$$

This relation plays an important role in the construction of the fast algorithms below. Firstly, we change Step II of Algorithm 2.1 as

$$\left\{ \begin{array}{l} d_1^{k+1} := \arg \min_{d_1} \left\{ |d_1|_{L^1(\Omega)} + \frac{\tau_1}{2} \|d_1 - \Lambda_1 u^k - b_1^k\|_{L^2(\Omega)}^2 \right\}, \\ d_2^{k+1} := \arg \min_{d_2} \left\{ |d_2|_{L^1(\Omega)} + \frac{\tau_2}{2} \|d_2 - \Lambda_2 u^k - b_2^k\|_{L^2(\Omega)}^2 \right\}, \\ b_1^{k+1} := b_1^k + \Lambda_1 u^k - d_1^{k+1}, \\ b_2^{k+1} := b_2^k + \Lambda_2 u^k - d_2^{k+1}, \\ u^{k+1} := \arg \min_u \left\{ \frac{\gamma}{2} \|u - f\|_{L^2(\Omega)}^2 + \frac{\tau_1}{2} \|d_1^{k+1} - \Lambda_1 u - b_1^k\|_{L^2(\Omega)}^2 + \frac{\tau_2}{2} \|d_2^{k+1} - \Lambda_2 u - b_2^k\|_{L^2(\Omega)}^2 \right\}. \end{array} \right.$$

Using the optimality conditions yields

$$\begin{cases} d_1^{k+1} = S_{\tau_1}(b_1^k + \Lambda_1 u^k), \\ d_2^{k+1} = S_{\tau_2}(b_2^k + \Lambda_2 u^k), \\ b_1^{k+1} = b_1^k + \Lambda_1 u^k - d_1^{k+1}, \\ b_2^{k+1} = b_2^k + \Lambda_2 u^k - d_2^{k+1}, \\ u^{k+1} = f - \frac{\tau_1}{\gamma} \Lambda_1^*(\Lambda_1 u^{k+1} - d_1^{k+1} + b_1^k) - \frac{\tau_2}{\gamma} \Lambda_2^*(\Lambda_2 u^{k+1} - d_2^{k+1} + b_2^k). \end{cases} \quad (2.9)$$

Notice that the last step requires to solve a PDE for u^{k+1} . To utilize the identity (2.8), we apply a Jacobi iteration to solve (2.9):

$$u^{k+1} = f - \frac{\tau_1}{\gamma} \Lambda_1^*(\Lambda_1 u^k - d_1^{k+1} + b_1^k) - \frac{\tau_2}{\gamma} \Lambda_2^*(\Lambda_2 u^k - d_2^{k+1} + b_2^k). \quad (2.10)$$

Accordingly, it follows from (2.8) that (2.9) can be reformulated as a compact form

$$\begin{cases} b_1^{k+1} = P_{B_{\tau_1}}(b_1^k + \Lambda_1 u^k), \\ b_2^{k+1} = P_{B_{\tau_2}}(b_2^k + \Lambda_2 u^k), \\ u^{k+1} = f - \frac{\tau_1}{\gamma} \Lambda_1^* b_1^{k+1} - \frac{\tau_2}{\gamma} \Lambda_2^* b_2^{k+1}. \end{cases} \quad (2.11)$$

This leads to the new projection method for (2.1).

Algorithm 2.2. Projection method (PM) for solving (2.1)

- (I) Set $u^0 = f$, $b_1^0 = 0$, $b_2^0 = 0$ and $k = 0$;
- (II) Compute $(u^{k+1}, b_1^{k+1}, b_2^{k+1})$ by (2.11);
- (III) If the stop criterion is not satisfied, set $k := k + 1$ and go to Step II.

Remark 2.1. Although a Jacobi iteration is used to approximate (2.9), we find that several iterations provide very satisfactory results and in most applications, only one iteration may work well. Notice that Step II in Algorithm 2.2 is easy to compute as it only involves a projection operator and an operator multiplication, so no need to solve PDEs. We see that this is based on replacing $\Lambda_i u^k - d_i^{k+1} + b_i^k$ in Algorithm 2.1 by b_i^{k+1} for $i = 1, 2$. In fact, it was also observed by Zhao and Jia [24]. However, their method did not give the relation between the split Bregman method and the projection method. On the other hand, by comparing Algorithm 2.2 with the gradient descent projection algorithm as in [9], it is not difficult to find that $\frac{1}{\gamma} b^*$ as a limit generated by Algorithm 2.2 can be regarded as the solution of the dual problem of (2.1). \square

We next consider the convergence of Algorithm 2.2. Let X and Y be Hilbert spaces. Recall that the Lipschitz condition $L(\Lambda)$ of the operator $\Lambda : X \rightarrow Y$ is defined by

$$\|\Lambda u - \Lambda v\|_{L^2(\Omega)} \leq L(\Lambda) \|u - v\|_{L^2(\Omega)}$$

for every $u, v \in X$.

Theorem 2.3. Assume that $\tau_1 \in (0, \frac{1}{\gamma L(\Lambda_1 \Lambda_1^*)})$ and $\tau_2 \in (0, \frac{1}{\gamma L(\Lambda_2 \Lambda_2^*)})$, then the sequence $\{(u^k, b_1^k, b_2^k)\}$ generated by Algorithm 2.2 converges to a limit point (u^*, b_1^*, b_2^*) . Furthermore, u^* is the solution of (1.5).

The proof of this theorem is given in Appendix C.

It is known that the above projection scheme converges like $O(k^{-1})$. Recently, an improved version of the first-order method (called FISTA) with the convergence rate $O(k^{-2})$ proposed by Beck and Teboulle [4] has been extended to solve the ROF model [36]. This method can be actually viewed as an extension of the classical gradient algorithm. Motivated by this, we improve the above projection method as follows.

Algorithm 2.3. Fast projection method (FPM) for solving (2.1):

- (I) Set $u^0 = f$, $b_1^0 = 0$, $b_2^0 = 0$, $d_1^0 = 0$, $d_2^0 = 0$, $t_0 = 1$ and $k = 0$;
 (II) Compute $(u^{k+1}, b_1^{k+1}, b_2^{k+1})$ by
- $$d_1^k = P_{B_{\tau_1}}(b_1^k + \Lambda_1 u^k),$$
- $$d_2^k = P_{B_{\tau_2}}(b_2^k + \Lambda_2 u^k),$$
- $$t_{k+1} = \frac{1}{2}(1 + \sqrt{1 + 4t_k^2}),$$
- $$b_1^{k+1} = d_1^k + \frac{t_k - 1}{t_{k+1}}(d_1^k - d_1^{k-1}),$$
- $$b_2^{k+1} = d_2^k + \frac{t_k - 1}{t_{k+1}}(d_2^k - d_2^{k-1}),$$
- $$u^{k+1} = f - \frac{\tau_1}{\gamma} \Lambda_1^* b_1^{k+1} - \frac{\tau_2}{\gamma} \Lambda_2^* b_2^{k+1};$$
- (III) If the stop criterion is not met, set $k := k + 1$ and go to Step II.

Remark 2.2. Difference to Algorithm 2.3, there was also an improvement for Algorithm 2.2 based on the relaxed iterative method in [25]. Though this relaxed version was observed to be faster than Algorithm 2.3, there are not some analysis of its convergence rate. So here we propose Algorithm 2.3 to improve the convergence rate of Algorithm 2.2 and make it obtain $O(k^{-2})$. \square

In order to prove the convergence of Algorithm 2.3, we first consider the dual formulation of (1.5).

Theorem 2.4. The solution of (1.5) is given by

$$u = f - \frac{1}{\gamma}(\Lambda_1^* p_1 + \Lambda_2^* p_2),$$

where $p = (p_1, p_2)$ satisfies

$$\min_{\|p\|_{L^\infty} \leq 1} \|\Lambda_1 p_1 + \Lambda_2 p_2 - \gamma f\|_{L^2(\Omega)}^2. \quad (2.12)$$

The proof of this theorem is provided in Appendix D.

Setting $F(p) = \|\Lambda_1 p_1 + \Lambda_2 p_2 - \gamma f\|_{L^2(\Omega)}^2$ and

$$g(p) = \mathcal{X}_K(p) = \begin{cases} 0 & \text{if } p \in K = \{p : \|p\|_{L^1} \leq 1\}, \\ +\infty & \text{otherwise,} \end{cases}$$

we can rewrite (2.12) as

$$\min_p \{F(p) + g(p)\}. \quad (2.13)$$

It is obvious that $F(p)$ is a smooth convex function of $C^{1,1}$ -type and satisfies the following Lipschitz condition

$$\|\nabla F(p) - \nabla F(q)\|_{L^2(\Omega)} \leq L\|p - q\|_{L^2(\Omega)}, \quad \forall p, q \in \mathbb{R}^2,$$

where L is the Lipschitz constant. Therefore, based on the problem (2.13), we can deduce that the problem (1.5) falls into the framework of the model proposed by Beck and Teboulle [4]. This implies that Algorithm 2.3 can be looked at as the fast iterative shrinkage-thresholding algorithm (FISTA), so we have the following result.

Theorem 2.5. *Assume that (u^*, b_1^*, b_2^*) is the optimal solution of the problem (2.1). If $\tau_1 \in (0, \frac{1}{\gamma L(\Lambda_1 \Lambda_1^*)})$ and $\tau_2 \in (0, \frac{1}{\gamma L(\Lambda_2 \Lambda_2^*)})$, then the sequence $\{(u^k, b_1^k, b_2^k)\}$ generated by Algorithm 2.3 converges to (u^*, b_1^*, b_2^*) . Furthermore, Algorithm 2.3 has a convergence rate $O(k^{-2})$ and u^* is the solution of (1.5).*

3. Implementation on the Anisotropic LLT Model

As an important application, we apply the algorithms introduced in the previous section to the anisotropic LLT model (1.4). We first introduce some notations. Assume that Ω is a square domain in \mathbb{R}^2 and Ω is divided uniformly into N^2 sub-square domains $\Omega_{i,j}$ ($i, j = 1, \dots, N$). Let $x_{i,j} = (i, j)$ be the left-down vertex of $\Omega_{i,j}$. Let $u_{i,j}$ be the approximation of u at $x_{i,j}$. Define the differential operators as in [27]:

$$\begin{aligned} D_x u_{i,j} &:= [u_{i+1,j} - u_{i,j}], \\ D_{xx} u_{i,j} &:= [D_x(u_{i,j}) - D_x(u_{i-1,j})], \\ D_y u_{i,j} &:= [u_{i,j+1} - u_{i,j}], \\ D_{yy} u_{i,j} &:= [D_x(u_{i,j}) - D_x(u_{i,j-1})]. \end{aligned}$$

Setting $\Lambda_1 = D_{xx}$ and $\Lambda_2 = D_{yy}$, the anisotropic LLT model (1.4) becomes a special case of (1.5).

Algorithm 3.1. The split Bregman method (SBM) for solving the Anisotropic LLT Model (1.4):

(I) Initialization: Set $u^0 = f$, $d_{xx}^0 = d_{yy}^0 = b_{xx}^0 = b_{yy}^0 = 0$. Let $k := 0$;

(II) Compute $(u^{k+1}, d_{xx}^{k+1}, d_{yy}^{k+1}, b_{xx}^{k+1}, b_{yy}^{k+1})$ by

$$u^{k+1} = G^k(\beta, \tau_1, \tau_2, u^k, d_{xx}^k, d_{yy}^k, b_{xx}^k, b_{yy}^k, f),$$

$$d_{xx}^{k+1} = \text{shrink}(D_{xx}u^{k+1} - b_{xx}^k, \tau^{-1}),$$

$$d_{yy}^{k+1} = \text{shrink}(D_{yy}u^{k+1} - b_{yy}^k, \tau^{-1}),$$

$$b_{xx}^{k+1} = b_{xx}^k + D_{xx}u^{k+1} - d_{xx}^{k+1},$$

$$b_{yy}^{k+1} = b_{yy}^k + D_{yy}u^{k+1} - d_{yy}^{k+1},$$

where $G^k(\cdot)$ for $i, j = 1, 2, \dots, N$ takes for form

$$\begin{aligned} G_{i,j}^k(\cdot) = & \frac{\tau_1}{\beta + 6\tau_1 + 6\tau_2} (-u_{i+2,j}^k + 4u_{i+1,j}^k + 4u_{i-1,j}^k - u_{i-2,j}^k + d_{xx,i-1,j}^k \\ & - 2d_{xx,i,j}^k + d_{xx,i+1,j}^k - b_{xx,i-1,j}^k + 2b_{xx,i,j}^k - b_{xx,i+1,j}^k) \\ & + \frac{\tau_2}{\beta + 6\tau_1 + 6\tau_2} (-u_{i,j+2}^k + 4u_{i,j+1}^k + 4u_{i,j-1}^k - u_{i,j-2}^k + d_{yy,i,j-1}^k - 2d_{yy,i,j}^k \\ & + d_{yy,i,j+1}^k - b_{yy,i,j-1}^k + 2b_{yy,i,j}^k - b_{yy,i,j+1}^k) + \frac{\beta}{\beta + 6\tau_1 + 6\tau_2} f_{i,j}; \end{aligned}$$

(III) If the stopping criterion is not satisfied, the set $k := k + 1$ and go to Step II.

In fact, the first term of Step II in Algorithm 3.1 can be deduced from the first term of Step II in Algorithm 2.1 by using the Gauss-Seidel iteration to solve the optimality condition for u , which can be written as

$$(\beta I + \tau_1 D_{xx}^T D_{xx} + \tau_2 D_{yy}^T D_{yy})u^{k+1} = \beta f + \tau_1 D_{xx}^T (d_1^k - b_1^k) + \tau_2 D_{yy}^T (d_2^k - b_2^k).$$

To avoid solving equations, we apply the projection method to solve the anisotropic LLT model (1.4) as follows.

Algorithm 3.2. Projection Method (PM) for solving the Anisotropic LLT Model (1.4)

(I) Initialization: Set $u^1 = f$, $\bar{d}_{xx}^0 = \bar{d}_{yy}^0$. Let $k := 1$;

(II) Compute $(u^{k+1}, \bar{d}_{xx}^k, \bar{d}_{yy}^k)$ by

$$\bar{d}_{xx}^k := P_{B_{\tau_1}}(D_{xx}u^k + \bar{d}_{xx}^{k-1}),$$

$$\bar{d}_{yy}^k := P_{B_{\tau_2}}(D_{yy}u^k + \bar{d}_{yy}^{k-1}),$$

$$u^{k+1} := f - \frac{\tau_1}{\beta} D_{xx}^T \bar{d}_{xx}^k - \frac{\tau_2}{\beta} D_{yy}^T \bar{d}_{yy}^k;$$

(III) If the stopping criterion not satisfied, set $k := k + 1$ and go to Step II.

In contrast to Algorithm 3.1, we can directly obtain u^{k+1} by using the variables \tilde{d}_{xx}^k and \tilde{d}_{yy}^k in Algorithm 3.2. Moreover, the auxiliary variables b_{xx}^{k+1} and b_{yy}^{k+1} are not included in Algorithm 3.2, so Algorithm 3.2 is more economic than Algorithm 3.1.

Applying Algorithm 2.3 to the anisotropic LLT model leads to the following algorithm.

Algorithm 3.3. Fast Projection Method(FPM) for solving the Anisotropic LLT Model (1.4).

- (I) Initialization: Set $u^1 = f$, $\tilde{d}_{xx}^0 = \tilde{d}_{yy}^0$, $t_0 = 1$. Let $k:=1$;
- (II) Compute $(u^{k+1}, \tilde{d}_{xx}^k, \tilde{d}_{yy}^k)$ by

$$\tilde{d}_{xx}^k := P_{B_{\tau_1}}(D_{xx}u^k + \tilde{d}_{xx}^{k-1}),$$

$$\tilde{d}_{yy}^k := P_{B_{\tau_2}}(D_{yy}u^k + \tilde{d}_{yy}^{k-1}),$$

$$t_{k+1} := \frac{1}{2}(1 + \sqrt{1 + 4t_k^2}),$$

$$b_1^{k+1} := \tilde{d}_1^k + \frac{t_k - 1}{t_{k+1}}(\tilde{d}_1^k - \tilde{d}_1^{k-1}),$$

$$b_2^{k+1} := \tilde{d}_2^k + \frac{t_k - 1}{t_{k+1}}(\tilde{d}_2^k - \tilde{d}_2^{k-1}),$$

$$u^{k+1} := f - \frac{\tau_1}{\beta}D_{xx}^T b_1^{k+1} - \frac{\tau_2}{\beta}D_{yy}^T b_2^{k+1};$$
- (III) If the stopping criterion not satisfied, set $k := k + 1$ and go to (II).

We point out that the second-order differential operators D_{xx} and D_{yy} satisfy

$$\|D_{xx}u\|_{\ell^2} \leq 16\|u\|_{\ell^2} \quad \text{and} \quad \|D_{yy}u\|_{\ell^2} \leq 16\|u\|_{\ell^2}.$$

Therefore, we derive from Theorem 2.5 the following convergence results.

Corollary 3.1. For the anisotropic LLT model, then the following assertions hold.

- (1) The sequence $\{(u^k, d_{xx}^k, d_{yy}^k, b_{xx}^k, b_{yy}^k)\}$ generated by Algorithm 3.1 is convergent.
- (2) Assume that $\tau_1 \in (0, \frac{1}{16\beta})$ and $\tau_2 \in (0, \frac{1}{16\beta})$. Then the sequence $\{(u^k, \tilde{d}_{xx}^k, \tilde{d}_{yy}^k)\}$ generated by Algorithm 3.2 is convergent.
- (3) Assume that $\tau_1 \in (0, \frac{1}{16\beta})$ and $\tau_2 \in (0, \frac{1}{16\beta})$. Then the sequence $\{(u^k, \tilde{d}_{xx}^k, \tilde{d}_{yy}^k)\}$ generated by Algorithm 3.3 is convergent.

Furthermore, if u^* is the limit point corresponding to the iteration sequence for the above three algorithms, then u^* is the solution of the anisotropic LLT model (1.4).

4. Numerical Results and Discussions

This section is devoted to numerical experiments and a comparison study with an aim to show the performance of the proposed algorithms. We shall see that the use of

anisotropic LLT model can considerably reduce the staircase effect, compared with the ROF model. We shall demonstrate that the split Bregman-based method with a new projection scheme is very efficient for the anisotropic LLT model.

For comparison, we recall the dual method in Steidl [38] for (1.4), as summarized below.

Algorithm 4.1. Dual Method (DM) for solving the Anisotropic LLT Model (1.4).

- (I) Set $u^0 = f$, and $p_{xx}^0 = p_{yy}^0 = 0$;
 (II) While the stop criterion is not satisfied, compute the following iteration:

$$\begin{cases} u^k = f - \frac{1}{\beta}(D_{xx}p_{xx}^k + D_{yy}p_{yy}^k), \\ p_{xx}^{k+1} = \frac{p_{xx}^k + \frac{\tau}{\beta}(D_{xx}u^k)}{1 + \frac{\tau}{\beta}|D_{xx}u^k|}, \\ p_{yy}^{k+1} = \frac{p_{yy}^k + \frac{\tau}{\beta}(D_{yy}u^k)}{1 + \frac{\tau}{\beta}|D_{yy}u^k|}, \end{cases}$$

for $k = 1$ to N . Here τ is a positive parameter.

As in Chambolle [10], it is easy to deduce that the above DM is convergent when $\tau < \frac{1}{32}$. In order to measure the quality of the restored image, we introduce the signal to noise ratio (SNR) and the mean squared error (MSE) defined by

$$SNR = 10 \cdot \log_{10} \left(\frac{\int_{\Omega} (u - \bar{u})^2 d\Omega}{\int_{\Omega} (\eta - \bar{\eta})^2 d\Omega} \right) \text{ and } MSE = \frac{1}{|\Omega|} \int_{\Omega} (u - u_0)^2 d\Omega,$$

where $|\Omega| = \int_{\Omega} d\Omega$, $\bar{u} = \frac{1}{|\Omega|} \int_{\Omega} u d\Omega$ and $\bar{\eta} = \frac{1}{|\Omega|} \int_{\Omega} \eta d\Omega$, $\eta = u - u_0$ denotes the noise. All results are generated by using MATLAB(R2009a), on a PC with an Intel Core i5 M520 2.40 GHz processor, with 4GB of RAM. Furthermore, it is noteworthy that in the following computation, we take $\tau = \tau_1 = \tau_2$ and choice the suitable values of τ and β by the trials.

Example 1: We process a synthetic image with 128×128 pixels to show that the anisotropic LLT model can overcome the staircase effect. The noisy image is obtained by adding a Gaussian white noise with the standard variance $\sigma = 15$. The restored images in Fig. 1 are obtained by using the dual algorithm for the anisotropic ROF model (1.3) as did by Chambolle [9] and Algorithm 4.1 for the anisotropic LLT model (1.4), respectively. The parameters are chosen with $\lambda_2 = 0.08$, $t = 0.125$ for the anisotropic ROF model and $\beta = 0.137$, $\tau = 0.01$ for the anisotropic LLT model. Both of the algorithms will stop after 500 iterations. As we can see from Fig. 1(c), the image restored by the ROF model has staircase effect. However, the image restored by the LLT model looks more natural as shown in Fig. 1(d). In order to make a better comparison, we consider a profile line in Fig. 2(a) which station is shown in Fig. 1(a). From Fig. 2, it is easy to find that the image restored by the LLT model has much less staircase effect and looks more natural again.

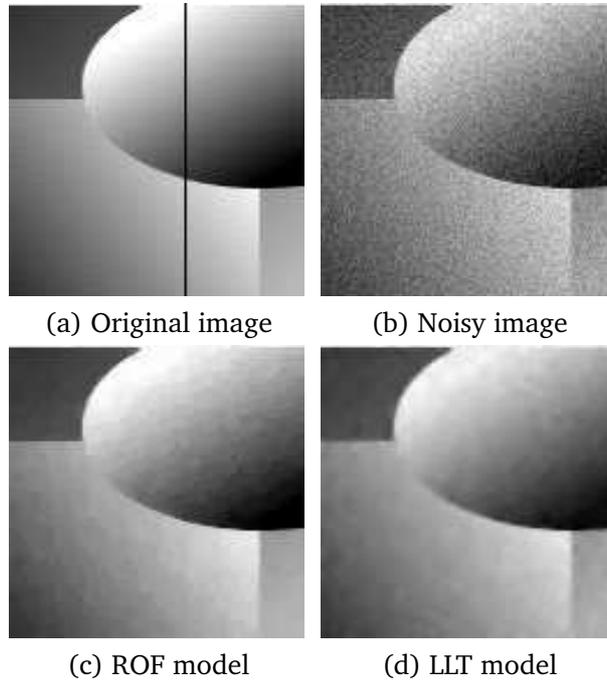


Figure 1: The related images: (b) $SNR = 12.1869$; (c) $SNR = 22.7887$; (d) $SNR = 22.6160$.

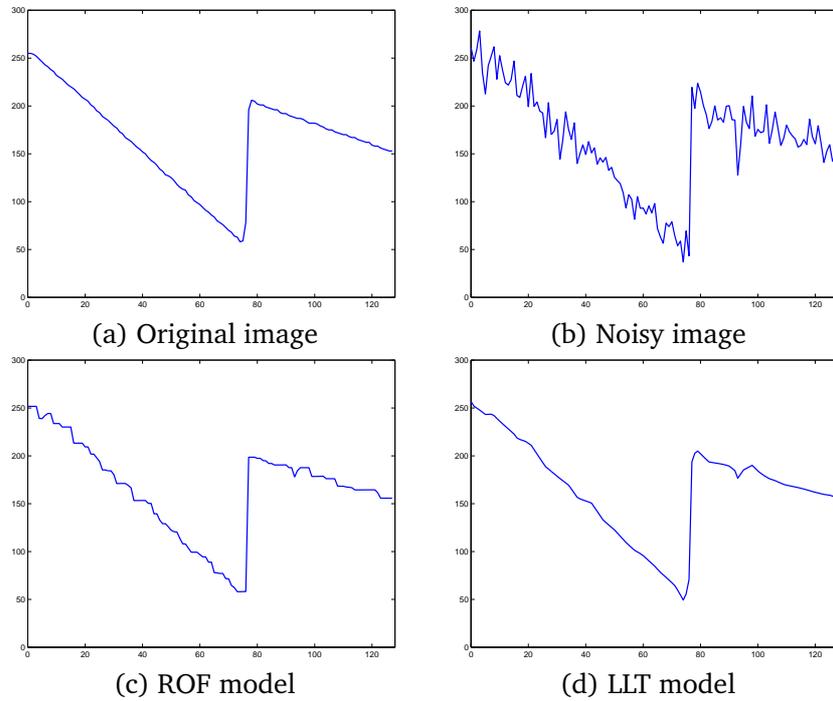


Figure 2: The profiles of related images in Fig. 1.

Table 1: The related data of Example 2.

Method	β	τ	Time(s)	MSE	SNR
DM	0.35	0.01	2.2304	34.0610	18.9826
SBM	0.35	0.005	3.1356	34.0207	18.9862
PM	0.35	0.0167	1.0452	34.2403	18.9598
FPM	0.35	0.0167	1.4196	34.1533	18.9705

Example 2: In this example, we compare the speed of the three algorithms in Section 3 and Algorithm 4.1. We consider the noisy Lena image contaminated by a Gaussian white noise with the standard variance $\sigma = 10$. The related parameters, the computations time, MSE, and SNR achieved by each of algorithms are arranged in Table 1. After the algorithms are implemented about 100 iterations, we output the restored images. We notice from Table 1 that the images restored by these four algorithms almost have the same MSE and SNR. However, it is not difficult to find that the curves of MSE and SNR in Fig. 4 generated by FPM obviously change faster than those generated by other methods, which implies that FPM outperforms other methods. On the other hand, we should also

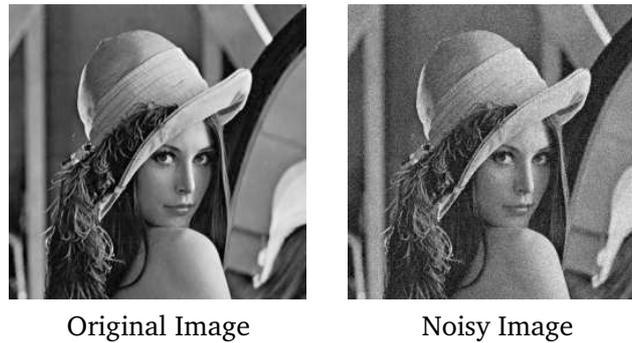
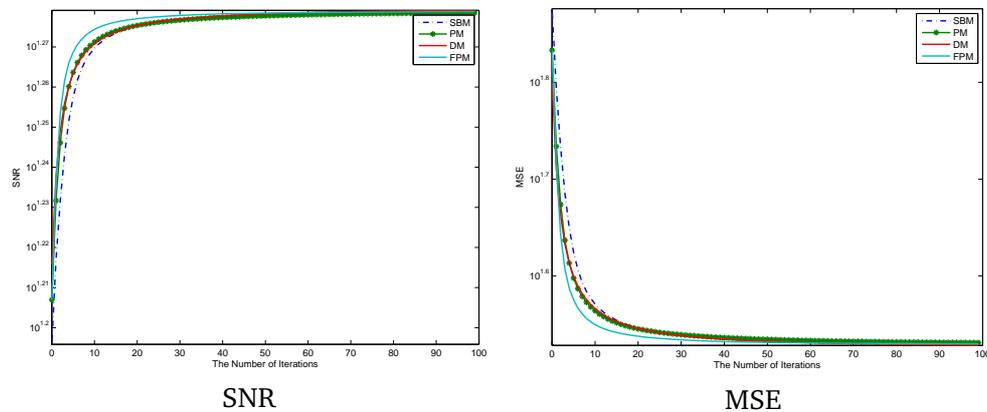
Figure 3: The related images: (b) $SNR = 14.3766$ and $MSE = 99.8160$.

Figure 4: The related curves.

notice the computation times. As we expect, Algorithms 3.2 and 3.3 spend much less the computation time. In fact, Algorithms 3.2 (PM) and 3.3 (FPM) only require 33.3% and 45.3% computation time of Algorithm 3.1 (SBM), respectively.

Example 3: To analyze the convergence of the proposed algorithms in Section 3, we consider the 256×256 Peppers image as the test image. The noisy image is contaminated by a Gaussian white noise with the standard variance $\sigma = 20$. Here, we take $\beta = 0.135$ and $\tau = 0.034$. Algorithms are terminated after 1000 iterations. At this time, we have $SNR_{SBM} = 15.2335$, $SNR_{PM} = 15.2791$ and $SNR_{FPM} = 15.2794$ for restored images.

In order to understand the change of energy function (1.4), we plot energy function curves corresponding to these three methods for the first 100 iterations. From Fig. 6(a), we observe that the energy function (1.4) of FPM changes much faster than those of both SBM and PM. To compare the accuracy $F(u^k) - F(u^*)$ of these three methods, we also plot convergence curves of the accuracy in Fig. 6(b). It is clear that FPM is fastest and can reach greatest accuracies amongst these three methods. Furthermore, the accuracy obtained by PM at iteration 1000 only require about 500 iterations for FPM and 550 iterations for SBM. The above comparison shows that FPM is the best among three methods.

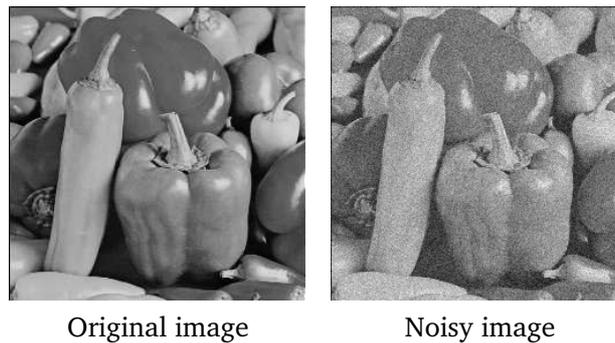


Figure 5: The related images: (b) $SNR = 8.0657$ and $MSE = 401.8418$.

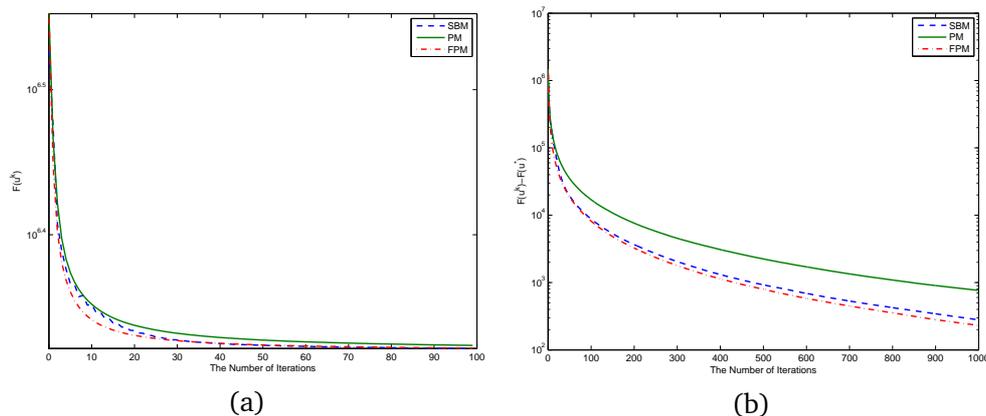


Figure 6: (a) The energy function curves of SBM, PM, and FPM for the first 100 iterations; (b) Accuracy $F(u^k) - F(u^*)$ of FPM, PM and SBM.

5. Concluding Remarks and Discussions

By using the relation between the projection operator and the shrink operator, we deduced a projection method from the split Bregman method to solve the minimization problems including two L^1 -regularization terms. The proposed projection method compared with the split Bregman method can avoid solving PDEs so that it saves computation time. Furthermore, in order to make the projection method achieve the convergence rate $O(k^{-2})$, we employed the strategy of the fast iterative shrinkage/thresholding algorithm (FISTA) to speed up the projection method. The convergence analysis of the proposed methods were provided. Furthermore, we applied these methods to solve the anisotropic LLT model and compared with various methods. The proposed algorithms have a great implication in other image processing tasks such as image deblurring, zooming and inpainting. On the other hand, the relaxed version in [25] encourages us to improve our proposed method and choose the related relaxed parameters in the farther work.

Acknowledgments

The first author would like to thank the Mathematical Imaging and Vision Group in Division of Mathematical Sciences at Nanyang Technological University in Singapore for the hospitality during the visit. The authors also appreciate Prof. Xue-Cheng Tai for many discussions. This research is supported by Singapore MOE Grant T207B2202, Singapore NRF2007IDM-IDM002-010, the NNSF of China (No. 60835004, 60872129).

Appendix

A. Proof of Theorem 2.1

Based on the assumption, since the sequence $\{(u^{k+1}, d_1^{k+1}, d_2^{k+1}, b_1^{k+1}, b_2^{k+1})\}$ is generated by Algorithm 2.1, then the optimality conditions of Step II in Algorithm 2.1 satisfy

$$\begin{cases} 0 = \gamma(u^{k+1} - f) - \tau_1 \Lambda_1^*(d_1^k - \Lambda_1 u^{k+1} - b_1^k) - \tau_2 \Lambda_2^*(d_2^k - \Lambda_2 u^{k+1} - b_2^k), & \text{(A.1a)} \\ 0 \in \partial(|d_1^{k+1}|) + \tau_1(d_1^{k+1} - \Lambda_1 u^{k+1} - b_1^k), & \text{(A.1b)} \\ 0 \in \partial(|d_2^{k+1}|) + \tau_2(d_2^{k+1} - \Lambda_2 u^{k+1} - b_2^k), & \text{(A.1c)} \\ b_1^{k+1} = b_1^k + \Lambda_1 u^{k+1} - d_1^{k+1}, & \text{(A.1d)} \\ b_2^{k+1} = b_2^k + \Lambda_2 u^{k+1} - d_2^{k+1}, & \text{(A.1e)} \end{cases}$$

where Λ_1^* and Λ_2^* are the adjoint operators of Λ_1 and Λ_2 , respectively. Set

$$\Lambda := \begin{pmatrix} \Lambda_1 \\ \Lambda_2 \end{pmatrix}, \quad d^k := \begin{pmatrix} d_1^k \\ d_2^k \end{pmatrix}, \quad b^k := \begin{pmatrix} b_1^k \\ b_2^k \end{pmatrix}, \quad \tau := \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix},$$

then (A.1a) can be rewritten as

$$\Lambda u^{k+1} = \frac{\tau}{\gamma} \Lambda \Lambda^*(g + d^k - \Lambda u^{k+1} - b^k),$$

where $\frac{\tau}{\gamma}\Lambda^*g := f$. Adding $g + d^k - \Lambda u^{k+1} - b^k$ to the two sides of the above equation, we can get

$$b^k - g - d^k = \left(I + \frac{\tau}{\gamma}\Lambda\Lambda^*\right)(b^k + \Lambda u^{k+1} - g - d^k), \quad (\text{A.2})$$

where I denotes the identity operator. Set $A(x) := \Lambda\Lambda^*(x - g)$ for $x \in \Omega$ and $\mu := \frac{\tau}{\gamma}$, then we can deduce that $(I + \mu A)x = (I + \mu\Lambda\Lambda^*)(x - g) + g$, which implies that

$$J_{\mu A}x := (I + \mu\Lambda\Lambda^*)^{-1}(x - g) + g$$

and $J_{\mu A}$ is firmly nonexpansive. Thus (A.2) can be rewritten as

$$\Lambda u^{k+1} + b^k = J_{\mu A}(b^k - d^k) + d^k. \quad (\text{A.3})$$

Let $v^k := b^k$ and $t^k := b^k + d^k$, we can rewrite (A.3) as

$$t^{k+1} = J_{\mu A}(2v^k - t^k) + (t^k - v^k). \quad (\text{A.4})$$

It is clear that (A.4) corresponds to the first iteration of (2.6) in Lemma 2.1.

Now we consider the second iteration of (2.6). It is clear to find that Eqs. (A.1a) and (A.1b) can be rewritten as

$$0 \in \partial(|d^{k+1}|) + \tau(d^{k+1} - \Lambda u^{k+1} - b^k).$$

Set $B^{-1}x = \partial(|x|)$, then B as a set-valued mapping implies that

$$By = x \text{ with } x = \begin{cases} [0, +\infty) & \text{if } y = 1, \\ 0 & \text{if } |y| \leq 1, \\ (-\infty, 0] & \text{if } y = -1 \end{cases}$$

for $y \in \Omega$. Hence we get

$$J_Bx = (I + B)^{-1}x = y \text{ with } y = \begin{cases} 1 & \text{if } x > 1, \\ x & \text{if } |x| \leq 1, \\ -1 & \text{if } x < -1. \end{cases}$$

Furthermore, we can obtain the following formula:

$$d^{k+1} \in \tau^{-1}B(\Lambda u^{k+1} + b^k - d^{k+1}).$$

By the fact that $b^{k+1} = b^k + \Lambda u^{k+1} - d^{k+1}$, the above formula can be rewritten as

$$b^k + \Lambda u^{k+1} \in \left(\frac{1}{\tau}B + I\right)b^{k+1}.$$

Set $v^{k+1} := b^{k+1}$ and $t^{k+1} := b^k + \Lambda u^{k+1}$, then we get

$$v^{k+1} = J_{\tau^{-1}B}(t^{k+1}). \quad (\text{A.5})$$

Combining (A.4) with (A.5) and based on the maximal monotone operators A and B , it follows from Lemma 2.1 and $\tau, \mu > 0$ that the assertion holds. \square

B. Proof of Theorem 2.2

Based on the assumption, if $(u^*, d_1^*, d_2^*, b_1^*, b_2^*)$ is the limit point of the sequence $\{(u^k, d_1^k, d_2^k, b_1^k, b_2^k)\}$ generated by Algorithm 2.1, it is easy to deduce that $(u^*, d_1^*, d_2^*, b_1^*, b_2^*)$ satisfies

$$\begin{cases} 0 = \gamma(u^* - f) - \tau_1 \Lambda_1^*(d_1^* - \Lambda_1 u^* - b_1^*) - \tau_2 \Lambda_2^*(d_2^* - \Lambda_2 u^* - b_2^*), \\ 0 \in \partial(|d_1^*|) + \tau_1(d_1^* - \Lambda_1 u^* - b_1^*), \\ 0 \in \partial(|d_2^*|) + \tau_2(d_2^* - \Lambda_2 u^* - b_2^*), \\ b_1^* = b_1^* + d_1^* - \Lambda_1 u^*, \\ b_2^* = b_2^* + d_2^* - \Lambda_2 u^*. \end{cases} \quad (\text{B.1})$$

Setting $b_1^* = b_1^* + d_1^* - \Lambda_1 u^*$ and $b_2^* = b_2^* + d_2^* - \Lambda_2 u^*$, we rewrite (B.1) as

$$\begin{cases} 0 = \gamma(u^* - f) + \tau_1 \Lambda_1^* b_1^* + \tau_2 \Lambda_2^* b_2^*, \\ 0 \in \partial(|\Lambda_1 u^*|) - \tau_1 b_1^*, \\ 0 \in \partial(|\Lambda_2 u^*|) - \tau_2 b_2^*. \end{cases}$$

By the fact that Λ_1 and Λ_2 are bounded linear operators, we can deduce that

$$0 \in \gamma(u^* - f) + \Lambda_1^* \partial(|\Lambda_1 u^*|) + \Lambda_2^* \partial(|\Lambda_2 u^*|),$$

which implies that u^* is the solution of (1.5). \square

C. Proof of Theorem 2.3

We first notice that Step II of Algorithm 2.2 can be rewritten as

$$\begin{cases} b_1^{k+1} = P_{B_{\tau_1}} \left((I - \frac{\tau_1}{\gamma} \Lambda_1 \Lambda_1^*) b_1^k + \Lambda_1 f - \frac{\tau_1}{\gamma} \Lambda_1 \Lambda_2^* b_2^k \right), \\ b_2^{k+1} = P_{B_{\tau_2}} \left((I - \frac{\tau_2}{\gamma} \Lambda_2 \Lambda_2^*) b_2^k + \Lambda_2 f - \frac{\tau_2}{\gamma} \Lambda_2 \Lambda_1^* b_1^k \right), \end{cases}$$

so we can deduce that the above iterative sequence $\{(b_1^k, b_2^k)\}$ converges to a fixed point (b_1^*, b_2^*) when $\tau_1 \in (0, \frac{1}{\gamma L(\Lambda_1 \Lambda_1^*)})$ and $\tau_2 \in (0, \frac{1}{\gamma L(\Lambda_2 \Lambda_2^*)})$ [3]. It follows from Step II of Algorithm 2.2 that

$$\begin{cases} b_1^* = P_{B_{\tau_1}}(b_1^* + \Lambda_1 u^*), \\ b_2^* = P_{B_{\tau_2}}(b_2^* + \Lambda_2 u^*), \\ u^* = f - \frac{\tau_1}{\gamma} \Lambda_1^* b_1^* - \frac{\tau_2}{\gamma} \Lambda_2^* b_2^*, \end{cases} \quad (\text{C.1})$$

where u^* is the limit of $\{u^k\}$. So it is not difficult to see that (C.1) can be reformulated as

$$\begin{cases} \Lambda_1^* b_1^* = \frac{1}{\tau_1} \partial(|\Lambda_1 u^*|), \\ \Lambda_2^* b_2^* = \frac{1}{\tau_2} \partial(|\Lambda_2 u^*|), \\ u^* = f - \frac{\tau_1}{\gamma} \Lambda_1^* b_1^* - \frac{\tau_2}{\gamma} \Lambda_2^* b_2^*, \end{cases}$$

which imply

$$u^* - f + \frac{1}{\gamma}(\partial(|\Lambda_1 u^*|) + \partial(|\Lambda_2 u^*|)) = 0.$$

Using the uniqueness of the solution, we can deduce that u^* is the solution of (1.5). \square

D. Proof of Theorem 2.4

Using the fact that

$$|x|_{L^1(\Omega)} = \max_{\|p\|_{L^\infty(\Omega)} \leq 1} \{(x, p)_{L^2(\Omega)}\},$$

it follows that (1.5) can be rewritten as

$$\min_u \max_{\|p\|_{L^\infty(\Omega)} \leq 1} \frac{\gamma}{2} \|u - f\|_{L^2(\Omega)}^2 + (\Lambda_1^* p_1, u)_{L^2(\Omega)} + (\Lambda_2^* p_2, u)_{L^2(\Omega)},$$

where Λ_i^* is the adjoint operator of Λ_i for $i = 1, 2$. Since the objective function is convex for u and concave for p , by interchanging the order of the minimum and maximum, we can get

$$\max_{\|p\|_{L^\infty(\Omega)} \leq 1} \min_u \frac{\gamma}{2} \|u - f\|_{L^2(\Omega)}^2 + (\Lambda_1^* p_1, u)_{L^2(\Omega)} + (\Lambda_2^* p_2, u)_{L^2(\Omega)}, \quad (\text{D.1})$$

which implies that u as the function of p_1 and p_2 satisfies

$$u = f - \frac{1}{\gamma}(\Lambda_1^* p_1 + \Lambda_2^* p_2). \quad (\text{D.2})$$

By replacing the variable u in (D.1) by the expression (D.2) and omitting the constant terms, it follows that p_1 and p_2 satisfy

$$\min_{\|p\|_{L^\infty(\Omega)} \leq 1} \|\Lambda_1 p_1 + \Lambda_2 p_2 - \gamma f\|_{L^2(\Omega)}^2.$$

\square

References

- [1] L. Ambrosio, N. Fusco, D. Pallara. Functions of Bounded Variation and Free Discontinuity Problems. Oxford University Press, 2000.
- [2] G. Aubert and P. Kornprobst. Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations. Springer-Verlag, New York, 2002.
- [3] J. Aujol. Some first-order algorithms for total variation based image restoration. Journal of Mathematical Imaging and Vision, 34(3)(2009), 307-327.
- [4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences, 2(1)(2009), 183-202.

- [5] P. Blomgern, T. Chan, P. Mulet, and C. Wong. Total variation image restoration: Numerical methods and extensions. In: Proceedings, IEEE International Conference on Image Processing, III(1997), 384-387.
- [6] C. Brune, A. Sawatzky, and M. Burger. Primal and dual Bregman methods with application to optical nanoscopy. *International Journal of Computer Vision*, 92(2)(2011), 211-229.
- [7] M. Burger, K. Frick, S. Osher, and O. Scherzer. Inverse total variation flow. *Multiscale Modeling Simulation*, 6(2)(2007), 366-395.
- [8] J. F. Cai, S. Osher, and Z. Shen. Split Bregman methods and frame based image restoration. *Multiscale Modeling Simulation*, 8(2)(2009), 337-369.
- [9] A. Chambolle. Total variation minimization and a class of binary MRF models. In *EMMCVPR 05*, volume 3757 of *Lecture Notes in Computer Sciences*, (2005), 136-152.
- [10] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2)(2004), 89-97.
- [11] A. Chambolle and P. L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(1997), 167-188.
- [12] T. Chan, G. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *Journal on Scientific Computing*, 20(1999), 1964-1977.
- [13] T. Chan, A. Marquina, and P. Mulet. High-order total variation-based image restoration. *SIAM Journal on Scientific Computing*, 22(2)(2000), 503-516.
- [14] T. Chan and J. Shen. *Image Processing and Analysis-Variational, PDE, wavelet, and stochastic methods*. SIAM Publisher, Philadelphia, 2005.
- [15] H. Z. Chen, J. P. Song, and X. C. Tai. A dual algorithm for minimization of the LLT model. *Advances in Computational Mathematics*, 31(1-3)(2009), 115-130.
- [16] K. Chen and X. C. Tai. A nonlinear multigrid method for total variation minimization from image restoration. *Journal of Scientific Computing*, 33(2)(2007), 115-138.
- [17] P. L. Combettes. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization*, 53(5-6)(2004), 475-504.
- [18] J. Darbon and M. Sigelle. A fast and exact algorithm for total variation minimization. *The second ICPRIA*, 3522(2005), 351-359.
- [19] J. Douglas and H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2)(1956), 421-439.
- [20] E. Esser, X. Zhang, and T. Chan. A general framework for a class of first order primal-dual algorithms for TV minimization. *SIAM Journal on Imaging Sciences*, 3(4)(2010), 1015-1046.
- [21] T. Goldstein, X. Bresson, and S. Osher. Geometric applications of the split Bregman method: segmentation and surface reconstruction. *Journal of Scientific Computing*, 45(1-3)(2010), 272-293.
- [22] T. Goldstein and S. Osher. The split Bregman for L^1 regularized problems. *SIAM Journal on Imaging Sciences*, 2(2)(2009), 323-343.
- [23] L. He, A. Marquina, and S. Osher. Blind deconvolution using TV regularization and Bregman iteration. *International Journal of Imaging Systems and Technology*, 15(1)(2005), 74-83.
- [24] R. Q. Jia and H. Zhao. A fast algorithm for the total variation model of image denoising. *Advances in Computational Mathematics*, 33(2)(2010), 231-241.
- [25] R. Q. Jia, H. Zhao, and W. Zhao. Relaxation methods for image denoising based on difference schemes. *Multiscale Modeling and Simulation*, 9(1)(2011), 355-372.
- [26] Y. Li and F. Santosa. A computational algorithm for minimizing total variation in image restoration. *IEEE Transactions on Image Processing*, 5(6)(1996), 987-995.
- [27] M. Lysaker, A. Lundervold, and X.-C. Tai. Noise removal using fourth-order partial differential

- equation with applications to medical magnetic resonance images in space and time. *IEEE Transactions on Image Processing*, 12(12)(2003), 1579-1590.
- [28] A. Marquina and S. Osher. Image super-resolution by TV-regularization and Bregman iteration. *Journal of Scientific Computing*, 37(3)(2008), 367-382.
 - [29] C. Micchelli, L.-X. Shen, and Y.-S. Xu. Proximity algorithms for image models: denoising. *Inverse Problems*, 27(4)(2011). (Doi: 10.1088/0266-5611/27/4/045009).
 - [30] M. K. Ng, L. Qi, Y. F. Yang, and Y. M. Huang. On semismooth Newton's methods for total variation minimization. *Journal of Mathematical Imaging and Vision*, 27(3)(2007), 265-276.
 - [31] 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)(2005), 460-489.
 - [32] Z. F. Pang and Y. F. Yang. Semismooth Newton's methods for minimization of the LLT model. *Inverse problems and Imaging*, 3(4)(2009), 677-691.
 - [33] N. Paragios, Y. Chen, and O. Faugeras. *Handbook of Mathematical Models in Computer Vision*. Springer, Heidelberg, 2005.
 - [34] G. Plonka and J. Ma. Curvelet-wavelet regularized split Bregman iteration for compressed sensing. *International Journal of Wavelets. Multiresolution and Information Processing*, (2011), Preprint.
 - [35] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1992), 259-268.
 - [36] S. Setzer. Splitting methods in image processing. PhD Dissertation, (2009).
 - [37] O. Scherzer. Denoising with higher order derivatives of bounded variation and an application to parameter estimation. *Computing*, 60(1998), 1-27.
 - [38] G. Steidl. A note on the dual treatment of higher order regularization functionals. *Computing*, 76(1-2)(2006), 135-148.
 - [39] G. Steidl and T. Teuber. Removing multiplicative noise by Douglas-Rachford splitting methods. *Journal of Mathematical Imaging and Vision*, 36(2)(2010), 168-184.
 - [40] X. C. Tai and C. Wu. Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model. *The 2nd international conference, SSVM, LNCS 5567(2009)*, 502-513.
 - [41] C. R. Vogel and M. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1)(1996), 227-238.
 - [42] 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)(2008), 248-272.
 - [43] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 1999.
 - [44] C. Wu and X.-C. Tai. Augmented Lagrangian method, dual methods and split-Bregman iterations for ROF, vectorial TV and higher order models. *SIAM Journal on Imaging Sciences*, 3(3)(2010) 300-339.
 - [45] 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)(2008), 143-168.
 - [46] Y. L. You and M. Kaveh. Fourth-order partial differential equation for noise removal. *IEEE Transactions on Image Processing*, 9(10)(2000), 1723-1730.