

# ALTERNATING DIRECTION ALGORITHMS FOR TOTAL VARIATION DECONVOLUTION IN IMAGE RECONSTRUCTION

MIN TAO\* AND JUNFENG YANG \*

**Abstract.** Image restoration and reconstruction from blurry and noisy observation is known to be ill-posed. To stabilize the recovery, total variation (TV) regularization was introduced by Rudin, Osher and Fatemi in [24], which has demonstrated superiority in preserving image edges. However, the nondifferentiability of TV makes the underlying optimization problems difficult to solve. In this paper, we propose to solve TV deconvolution problems by alternating direction method (ADM) — a variant of the classic augmented Lagrangian method for structured optimization. The main idea of our approach is to reformulate a TV problem as a linear equality constrained problem where the objective function is separable, and then minimize its augmented Lagrangian function using a Gauss-Seidel updates of both primal and dual variables. This ADM approach can be applied to both single- and multi-channel images with either Gaussian or impulsive noise, and permit cross-channel blurs when the underlying image has more than one channel. The per-iteration computational complexity of the algorithm is dominated by several fast Fourier transforms. We present extensive experimental results concerning different blurs and noise to compare with FTVd [34, 37, 38] — a state-of-the-art algorithm for TV image reconstruction, and the results indicate that the ADM approach is more stable and efficient.

**Key words.** Total variation, deconvolution, deblurring, alternating direction method, fast Fourier transform

**AMS subject classifications.** 68U10, 65J22, 65K10, 65T50, 90C25

**1. Introduction.** In this paper, we consider image restoration problem from blurry and noisy observation. Without loss of generality, we assume that the underlying image is grayscale and has a square domain. Let  $\bar{x} \in \mathbb{R}^{n^2}$  be an original  $n \times n$  image,  $K \in \mathbb{R}^{n^2 \times n^2}$  be a blurring (or convolution) operator,  $\omega \in \mathbb{R}^{n^2}$  be an additive noise, and  $f \in \mathbb{R}^{n^2}$  be an observation which satisfies the relationship

$$f = K\bar{x} + \omega. \tag{1.1}$$

Given  $K$ , our objective is to recover  $\bar{x}$  from  $f$ , which is known as deconvolution or deblurring. When  $K$  is the identity operator, recovering  $\bar{x}$  from  $f$  is referred as denoising.

It is well-known that recovering  $\bar{x}$  from  $f$  by directly inverting (1.1) is unstable and produces useless results because the solution is highly sensitive to the noise  $\omega$ . To stabilize the recovery of  $\bar{x}$ , one must utilize some prior information. In such a stabilization scheme, a common approach is to add a regularizer to certain data fidelity, resulting to the following reconstruction model

$$\min_x \Phi_{\text{reg}}(x) + \mu\Phi_{\text{fid}}(x, f), \tag{1.2}$$

where in the objective function,  $\Phi_{\text{reg}}(x)$  regularizes the solution by enforcing certain prior constraints,  $\Phi_{\text{fid}}(x, f)$  measures the violation of the relation between  $x$  and the observation  $f$ , and  $\mu$  is a positive parameter which is used to balance the two terms for minimization.

Traditional regularization techniques include the Tikhonov-like regularization [28], the TV regularization [24], both of which have been well studied in the literature, and others. A discrete Tikhonov-like regularization takes the form  $\Phi_{\text{reg}}(x) = \sum_i \sum_j \|(D^{(j)}x)_i\|^2$ , where  $D^{(j)}$ 's stand for a certain finite difference operator, the inner summation is taken over some index set, and the outer one is taken over all the pixels. Although the resulting minimization problems are relatively easy to solve, Tikhonov-like regularization tends to make

---

\*Department of Mathematics, Nanjing University, #22 Hankou Road, Nanjing, 210093, P.R. China (taom@njupt.edu.cn, jfyang@nju.edu.cn)

images overly smoothed and often fails to adequately preserve important image attributes such as sharp edges. In comparison, the total variation (TV) regularization, first introduced in [24] by Rudin, Osher and Fatemi for image denoising and then generalized to image deconvolution in [25], overcomes these drawbacks and has been shown both experimentally and theoretically to be suitable for preserving sharp edges. The discrete form of TV for a grayscale image  $x \in \mathbb{R}^{n^2}$  is given by

$$\text{TV}(x) = \sum_{i=1}^{n^2} \|D_i x\|_2, \quad (1.3)$$

where, for each  $i$ ,  $D_i x \in \mathbb{R}^2$  represents the first-order finite difference of  $x$  at pixel  $i$  in both horizontal and vertical directions, the quantity  $\|D_i x\|_2$  is the variation of  $x$  at pixel  $i$ , and the summation in (1.3) is taken over all pixels, which explains the name of TV. We note that the 2-norm can be replaced by the 1-norm in (1.3), in which case the resulting TV is an anisotropic discretization because the 1-norm is not rotation invariant. In contrast, the resulting TV is isotropic when the 2-norm is used. We emphasize that our approach applies to both the isotropic and the anisotropic TV deconvolution problems. For simplicity, we will treat only the isotropic case in detail because the treatment for the anisotropic case is completely analogous. We point out that, although the isotropic TV is often preferred over any anisotropic ones, both types of discretizations lead to the so-called staircasing effects; see e.g., [5, 23]. Besides Tikhonov and TV-like regularization, there are other well studied regularizer, such as the Mumford-Shah regularization [21] and its variants [1, 26].

The fidelity term in (1.2) is usually taken as a penalization on the difference between  $Kx$  and  $f$  measured in different norms. Two common choices of  $\Phi_{\text{fid}}(x)$  are  $\|Kx - f\|_2^2$  and  $\|Kx - f\|_1$ , which are, respectively, suitable for Gaussian and impulsive noise. Combining TV with the two types of fidelity, we get the two widely studied deconvolution models

$$\min_x \sum_{i=1}^{n^2} \|D_i x\|_2 + \frac{\mu}{2} \|Kx - f\|_2^2 \quad (1.4)$$

and

$$\min_x \sum_{i=1}^{n^2} \|D_i x\|_2 + \mu \|Kx - f\|_1. \quad (1.5)$$

In this paper, we will derive fast algorithms for both problems (1.4) and (1.5), which will be referred as TV/L<sup>2</sup> and TV/L<sup>1</sup>, respectively, and their multichannel extensions. In the rest of this section, we introduce the notation, give a brief review of existing methods for TV deconvolution problems, and then describe the organization of this paper.

**1.1. Notation.** We let superscript “ $\top$ ” be the transpose (conjugate transpose) operator for real (complex) quantities. In an iterative scheme, we let a superscript index be the counter of iteration number. The  $i$ th component of a vector  $v$  is denoted by  $v_i$ . The two first-order global finite difference operators in horizontal and vertical directions are, respectively, denoted by  $D^{(1)}$  and  $D^{(2)}$ , which are  $n^2$ -by- $n^2$  matrices. As used in (1.3),  $D_i \in \mathbb{R}^{2 \times n^2}$  is a two-row matrix formed by stacking the  $i$ th row of  $D^{(1)}$  on that of  $D^{(2)}$ . For vectors  $u$  and  $v$ , we let  $(u; v) \triangleq (u^\top, v^\top)^\top$ . Similarly, for matrices  $A_1$  and  $A_2$  with identical number of columns, we let  $(A_1; A_2) \triangleq (A_1^\top, A_2^\top)^\top$ . Without misleading, we let  $\sum_i$  be the sum taken over 1 to  $n^2$ . In the rest of this paper, we let  $\|\cdot\|$  be to the 2-norm. Additional notation will be introduced as the paper proceeds.

**1.2. A brief review of existing methods.** In this section, we review briefly some existing methods for solving TV deconvolution problems. These methods mainly concerned on the TV/L<sup>2</sup> problem (1.4) and most of them, if not all, can be generalized to the TV/L<sup>1</sup> problem (1.5).

The first class of methods are based on smoothing the TV term. Since TV is nonsmooth, which caused the main difficulty, a number of methods are based on smoothing it and solving resulting approximation problems. The TV/L<sup>2</sup> problem (1.4) is usually approximated by

$$\min_x \sum_i \sqrt{\|D_i x\|^2 + \epsilon} + \frac{\mu}{2} \|Kx - f\|^2, \quad (1.6)$$

where  $\epsilon > 0$  is a small constant. As such, ordinary methods for unconstrained optimization can be applied, such as gradient descent method which takes the form of

$$x^{k+1} = x^k - \tau_k g(x^k),$$

where  $\tau_k > 0$  is a steplength, and  $g(x)$  is the gradient of the smoothed objective function in (1.6). In the pioneer work [24], a time-marching scheme was used to solve a partial differential equation system, which in optimization point of view is equivalent to a constant steplength gradient descent method. This time-marching scheme suffers slow convergence especially when the iterate point approaches the solution set. Although certain line search can be incorporated to speedup convergence, it still converges slowly especially when  $\epsilon$  becomes small. Another well-known method for the smoothed TV/L<sup>2</sup> problem (1.6) is the linearized gradient method proposed in [33] for denoising and in [36] for deblurring. The idea of the linearized gradient method is to solve the Euler-Lagrangian equation via a fixed-point iteration. In each iteration of the linearized gradient method, a linear system needs to be solved, which becomes more and more difficult as  $\epsilon$  goes to 0, or as  $K$  becomes more ill-conditioned. As a result, it is not efficient especially for large and/or severely ill-conditioned problems. Furthermore, both the explicit time-marching scheme and the linearized gradient method are first-order methods and thus are linearly convergent at best. To overcome the linear convergence of first-order methods, the authors of [33] incorporated Newton method to solve (1.6). Indeed quadratic convergence was observed. However, the large per-iteration cost of Newton method prevents its practical applicability, especially to large sized images. Moreover, as  $\epsilon$  becomes smaller, the computation of Newton directions becomes more difficult.

Another class of algorithms for TV problems are the iterative shrinkage/thresholding (IST) algorithms, which are independently proposed and analyzed by several authors in different fields [8, 9, 32, 27]. In [3], Bioucas-Dias and Figueiredo introduced a two-step IST (TwIST) algorithm, which exhibits much faster convergence than the primal IST algorithm for ill-conditioned problems. Recently, Beck and Teboulle [2] present a new fast iterative shrinkage-thresholding algorithm (FISTA), which not only preserves the computational simplicity of IST but also has an optimal global rate of convergence. We note that IST-based algorithms for TV deconvolution requires to solve a denoising subproblem at each iteration and does not able to take advantage of problem structures.

Recently, a fast TV deconvolution algorithm called FTVd was proposed in [34] for the solution of (1.4). Since both  $K$  and the finite difference operators have structures, it is desirable to design an algorithm that can take advantage of problem structures. For this purpose, the authors of [34] first transformed (1.4) as an equivalent constrained problem of the form

$$\min_{x,y} \left\{ \sum_i \|\mathbf{y}_i\| + \frac{\mu}{2} \|Kx - f\|^2 : \mathbf{y}_i = D_i x, i = 1, \dots, n^2 \right\}, \quad (1.7)$$

where, for each  $i$ ,  $\mathbf{y}_i \in \mathbb{R}^2$  is an auxiliary vector, and considered solution algorithms for (1.7) that are capable of utilizing convolution structures. We note that in (1.7) the objective function is separable and the constraints are linear. For convenience, we let  $y = (y_1; y_2) \in \mathbb{R}^{2n^2}$ , where  $y_1$  and  $y_2$  are vectors of length

$n^2$  satisfying  $((y_1)_i; (y_2)_i) = \mathbf{y}_i \in \mathbb{R}^2$  for  $i = 1, \dots, n^2$ . Then, the classical quadratic penalty method was applied to (1.7) in [34], which gives the following formulation

$$\min_{x,y} \sum_i \left( \|\mathbf{y}_i\| + \frac{\beta}{2} \|\mathbf{y}_i - D_i x\|^2 \right) + \frac{\mu}{2} \|Kx - f\|^2, \quad (1.8)$$

where  $\beta \gg 0$  is a penalty parameter. Finally, an alternating minimization with respect to  $x$  and  $y$  was applied to (1.8). Since the objective function in (1.8) is separable with respect to each  $\mathbf{y}_i$ , and both  $K$  and the finite different operators are convolution matrices, both subproblems can be solved easily and exactly by either simple shrinkage or fast Fourier transforms (FFTs). Furthermore, the overall convergence of FTVd is accelerated by heuristic continuation on  $\beta$ . Experimental results provided in [34] show that FTVd converges much faster than those algorithms that can not make use of problem structures.

From optimization theory, the solution of (1.8) well approximates that of (1.4) only when  $\beta$  becomes large, which results to numerical difficulties. To avoid  $\beta$  going to infinity, the classical augmented Lagrangian method [20, 31] (ALM) was applied in [16] resulting the following iterative framework

$$\begin{cases} (x^{k+1}, y^{k+1}) \leftarrow \arg \min_{x,y} \mathcal{L}_{\mathcal{A}}(x, y, \lambda^k), \\ \lambda^{k+1} \leftarrow \lambda^k - \beta(y^{k+1} - D x^{k+1}), \end{cases} \quad (1.9)$$

where  $\mathcal{L}_{\mathcal{A}}(x, y, \lambda)$  is the augmented Lagrangian function of (1.7) defined by

$$\mathcal{L}_{\mathcal{A}}(x, y, \lambda) \triangleq \sum_i \left( \|\mathbf{y}_i\| - \lambda_i^\top (\mathbf{y}_i - D_i x) + \frac{\beta}{2} \|\mathbf{y}_i - D_i x\|^2 \right) + \frac{\mu}{2} \|Kx - f\|^2, \quad (1.10)$$

where each  $\lambda_i$  is a vector in  $\mathbb{R}^2$  and  $\lambda \in \mathbb{R}^{2n^2}$ , similar to  $y$ , is a reordering of  $\lambda_i$ ,  $i = 1, 2, \dots, n^2$ . In [16], the authors derived (1.9) from the Bregman iterative method [30]. It is well-known that the presence and iterative updates of multiplier  $\lambda$  avoids  $\beta$  going to infinity and guarantees convergence of (1.9) to a solution of (1.7). The disadvantage of (1.9) is the exact minimization of  $\mathcal{L}_{\mathcal{A}}$  with respect to  $(x, y)$  at each iteration, which requires its own iteration. In this paper, we propose the use of alternating direction method (ADM) for the solution of (1.4), which is a variant of the ALM (1.9).

**1.3. Organization.** The rest of this paper is organized as follows. In Section 2, we describe the framework of ADM and discuss how to apply it to TV/L<sup>2</sup>. In this section, we also review some recent applications of the ADM. In Section 3, we present comparison results with FTVd [34] to demonstrate the effectiveness of the ADM. In Section 4, we extend the ADM approach to multichannel image reconstruction and TV/L<sup>1</sup> problems along with comparison results with FTVd. Finally, some concluding remarks are drawn in Section 5.

## 2. Alternating Direction Method for TV/L<sup>2</sup>.

**2.1. A General Framework of the ADM.** The basic idea of ADM goes back to the work of Glowinski and Marocco [13] and Gabay and Mercier [12]. Let  $\theta_1(\cdot)$  and  $\theta_2(\cdot)$  be convex functionals and  $A$  be a continuous linear operator. The authors of [12] considered minimizing an energy function of the form

$$\min_u \theta_1(u) + \theta_2(Au). \quad (2.1)$$

By introducing an auxiliary variable  $v$ , problem (2.1) was equivalently transformed to

$$\min_{u,v} \{ \theta_1(u) + \theta_2(v) : Au = v \}, \quad (2.2)$$

which decoupled the difficulties relative to the functionals  $\theta_1(\cdot)$  and  $\theta_2(\cdot)$  from the possible ill-conditioning effects of the linear operator  $A$ . Then an alternating minimization was applied to the augmented Lagrangian function of (2.2):

$$\Theta(u, v, \lambda) = \theta_1(u) + \theta_2(v) - \lambda^\top (Au - v) + \frac{\beta}{2} \|Au - v\|^2, \quad (2.3)$$

resulting the following iterative scheme

$$\begin{cases} u^{k+1} \leftarrow \arg \min_u \Theta(u, v^k, \lambda^k), \\ v^{k+1} \leftarrow \arg \min_v \Theta(u^{k+1}, v, \lambda^k), \\ \lambda^{k+1} \leftarrow \lambda^k - \beta(Au^{k+1} - v^{k+1}). \end{cases} \quad (2.4)$$

Following the pioneer work [13, 12], the ADM was studied extensively in optimization and variational analysis. In [15], the algorithm is interpreted as the Douglas-Rachford splitting method [6] applied to a dual problem. The equivalence between the ADM and a proximal point method is shown in [7]. It is also studied in convex programming [11] and variational inequalities [18, 40, 29]. In the following, we apply the idea of (2.4) to TV/ $L^2$  problem (1.4), or equivalently (1.7).

**2.2. Applying the ADM to TV/ $L^2$ .** Let  $\mathcal{L}_{\mathcal{A}}(x, y, \lambda)$  be the augmented Lagrangian function of (1.7) which is defined in (1.10). Started at  $x = x^k$  and  $\lambda = \lambda^k$ , the alternating minimization idea of (2.4) applied to (1.7) yields the following iterative scheme

$$\begin{cases} y^{k+1} \leftarrow \arg \min_y \mathcal{L}_{\mathcal{A}}(x^k, y, \lambda^k), \\ x^{k+1} \leftarrow \arg \min_x \mathcal{L}_{\mathcal{A}}(x, y^{k+1}, \lambda^k), \\ \lambda^{k+1} \leftarrow \lambda^k - \beta(y^{k+1} - Dx^{k+1}). \end{cases} \quad (2.5)$$

It is easy to show that the minimization of  $\mathcal{L}_{\mathcal{A}}(x^k, y, \lambda^k)$  with respect to  $y$  is equivalent to  $n^2$  two-dimensional problems of the form

$$\min_{\mathbf{y}_i \in \mathbb{R}^2} \|\mathbf{y}_i\| + \frac{\beta}{2} \left\| \mathbf{y}_i - \left( D_i x^k + \frac{1}{\beta} (\lambda^k)_i \right) \right\|^2, \quad i = 1, 2, \dots, n^2. \quad (2.6)$$

According to [34, 37], the solution of (2.6) is given explicitly by the two-dimensional shrinkage

$$\mathbf{y}_i^{k+1} = \max \left\{ \left\| D_i x^k + \frac{1}{\beta} (\lambda^k)_i \right\| - \frac{1}{\beta}, 0 \right\} \frac{D_i x^k + \frac{1}{\beta} (\lambda^k)_i}{\|D_i x^k + \frac{1}{\beta} (\lambda^k)_i\|}, \quad i = 1, 2, \dots, n^2, \quad (2.7)$$

where  $0 \cdot (0/0) = 0$  is assumed. The computational cost of (2.7) is linear with respect to problem size. We note that, when the 1-norm is used in the definition of TV,  $\mathbf{y}_i^{k+1}$  is given by the simpler one-dimensional shrinkage

$$\mathbf{y}_i^{k+1} = \max \left\{ \left| D_i x^k + \frac{1}{\beta} (\lambda^k)_i \right| - \frac{1}{\beta}, 0 \right\} \circ \operatorname{sgn} \left( D_i x^k + \frac{1}{\beta} (\lambda^k)_i \right), \quad i = 1, 2, \dots, n^2,$$

where “ $\circ$ ” and “ $\operatorname{sgn}$ ” represent, respectively, point-wise product and the signum function, and all operations are implemented by componentwise. On the other hand, fixing  $\lambda = \lambda^k$  and  $y = y^{k+1}$  (recall that  $y$  is a reordering of  $\mathbf{y}_i$ ,  $i = 1, 2, \dots, n^2$ ), the minimization of  $\mathcal{L}_{\mathcal{A}}$  with respect to  $x$  is a least squares problem and the corresponding normal equations are

$$\left( D^\top D + \frac{\mu}{\beta} K^\top K \right) x = D^\top \left( y^{k+1} - \frac{1}{\beta} \lambda^k \right) + \frac{\mu}{\beta} K^\top f, \quad (2.8)$$

where  $D \triangleq (D^{(1)}; D^{(2)}) \in \mathbb{R}^{2n^2 \times n^2}$  is the global first-order finite difference operator. We follow the standard assumption of  $\mathcal{N}(K) \cap \mathcal{N}(D) = 0$ , where  $\mathcal{N}(\cdot)$  represents the null space of a matrix, which ensures the nonsingularity of the coefficient matrix in (2.8). Under the periodic boundary conditions for  $x$ , both  $D^\top D$  and  $K^\top K$  are block circulant matrices with circulant blocks, see e.g., [17, 22], and thus are diagonalizable by the 2D discrete Fourier transforms. As a result, equations (2.8) can be solved exactly in three steps. First, we compute the right-hand side vector in equations (2.8) and apply a forward FFT. Second, by componentwisely dividing the eigenvalues of  $D^\top D + \frac{\mu}{\beta} K^\top K$  (which needs to be computed only once) we get  $\mathcal{F}x^{k+1}$ , where  $\mathcal{F}$  represents the two dimensional discrete Fourier transform. Third, we apply the two dimensional inverse discrete Fourier transform to  $\mathcal{F}x^{k+1}$  to get the new iterate  $x^{k+1}$ . At each iteration, the main cost for solving (2.8) is two FFTs (including one inverse FFT), each at a cost of  $O(n^2 \log(n))$ . Finally, we update  $\lambda$  by

$$\lambda^{k+1} = \lambda^k - \beta(y^{k+1} - Dx^{k+1}). \quad (2.9)$$

We note that in both the augmented Lagrangian scheme (1.9) and the ADM scheme (2.5), a steplength  $\gamma > 0$  can be attached to the update of  $\lambda$ , i.e.,  $\lambda$  is updated as follows

$$\lambda^{k+1} = \lambda^k - \gamma\beta(y^{k+1} - Dx^{k+1}). \quad (2.10)$$

For the convergence of the ALM with such a relaxation parameter,  $\gamma \in (0, 2)$  is required. While the permitted range for convergence is shrank to  $(0, (\sqrt{5} + 1)/2)$  in the ADM scheme (2.5). The shrinkage in the permitted range of  $\gamma$  from  $(0, 2)$  in the ALM to  $(0, (\sqrt{5} + 1)/2)$  in the ADM is related to relaxing the exact minimization of  $\mathcal{L}_{\mathcal{A}}$  with respect to  $(x, y)$  to merely one round of alternating minimization. Under certain reasonable technical assumptions, convergence of the ADM framework (2.5) with a steplength  $\gamma \in (0, (\sqrt{5} + 1)/2)$  was established in [14, 15] in the context of variational inequality.

Now we are at a position to formally present the ADM approach for grayscale image restoration. We solve TV/L<sup>2</sup> problem (1.4), or equivalently (1.7), by an ADM scheme given below.

**ALGORITHM 1.** *Input*  $f, K, \mu > 0, \beta > 0$  and  $\lambda^0$ . *Initialize*  $x = f$  and  $\lambda = \lambda^0$ .

**While** “not converged”, **Do**

- 1) *Compute*  $y^{k+1}$  according to (2.7) for given  $(x^k, \lambda^k)$ .
- 2) *Compute*  $x^{k+1}$  via solving (2.8).
- 3) *Update*  $\lambda^{k+1}$  via (2.9).

**End Do**

It is feasible to terminate Algorithm 1 based on optimality conditions of (1.7). For simplicity, we terminated Algorithm (1) by relative change in  $x$  in all of our experiments, i.e.,

$$\frac{\|x^{k+1} - x^k\|}{\max\{\|x^k\|, 1\}} < \epsilon, \quad (2.11)$$

where  $\epsilon > 0$  is a given tolerance.

**2.3. Recent Applications of the ADM.** Recently, the ADM has been successfully applied to many optimization problems that arise from different applications. Based on the splitting technique for TV as used in (1.7), the ADM has been applied to magnetic resonance image reconstruction from partial frequencies in [39]. When we started the writing of this paper, we found a newly released preprint [10], which discussed the connections between FTVd [38], the split Bregman iteration [16], and the ADM. The author proposed using the ADM to deal with different image processing problems of both primal and dual forms. Also, in [42] the ADM has been applied to solve eight  $\ell_1$ -minimization problems that arise from sparse solution recovery and

compressive sensing. Lately, the ADM has also been applied to semi-definite programming in [35], matrix completion in [4], low-rank and sparse matrix decomposition in [41], and least squares covariance matrix problem in [19].

The versatility of the ADM has demonstrated its efficiency and robustness especially for large-scale problems which are beyond the size solvable by interior-point methods. The main contribution of this paper is to apply the ADM to TV-based deconvolution problems, including (1.4), (1.5) and their multichannel variants, and to provide extensive comparison results to demonstrate its efficiency. We will discuss the extensions to multichannel image deconvolution in Section 4.

**3. Numerical experiments.** In this section, we present numerical results to compare ADM<sup>1</sup> with FTVd.v3.0, which implements quadratic penalty method to (1.7) with continuation on  $\beta$  and has been shown to be highly efficient for solving TV deconvolution problems. For simplicity, in the following FTVd refers to FTVd.v3.0. We designed a set of experiments in terms of different image sizes, blurs and noise. We tested several images including Cameraman (256×256) and Lena (512×512), both of which have nice mixture of details, flat regions, shading areas, and textures. All experiments were performed under Windows XP and MATLAB v7.1 (R14) running on a desktop with an AMD Ultra X4 Processor 2.6GHZ and 1GB of memory.

**3.1. Parameter values.** As is usually done, we measured the quality of restoration by the signal-to-noise ratio (SNR), which is measured in decibel (dB) and defined by

$$\text{SNR}(x) \triangleq 10 * \log_{10} \frac{\|\bar{x} - \tilde{x}\|^2}{\|\bar{x} - x\|^2},$$

where  $\bar{x}$  is the original image and  $\tilde{x}$  is the mean intensity value of  $\bar{x}$ . In our experiments, we used  $\mu = 0.05/\text{std}^2$ , where  $\text{std}$  is the standard deviation of the additive Gaussian noise  $\omega$  in (1.1). This formula is based on the observation that  $\mu$  should be inversely proportional to the variance of noise, while the constant 0.05 was determined empirically so that the restored images had reasonable SNR. For tests on impulsive noise, we determined  $\mu$  in (1.5) such that the recovered images have reasonable SNR values.

As stated before, a substantial advantage of the ADM compared with FTVd is that parameter  $\beta$  no longer needs to go to infinity for the convergence to (1.7). As for  $\gamma$  in (2.10), we note that although it can vary in  $(0, \frac{\sqrt{5}+1}{2})$  without hurting convergence of the ADM, this relaxation parameter does not notably affect convergence speed. Therefore, we simply set  $\gamma = 1.618$  in all the tests. In order to determine a suitable  $\beta$  in (1.10), we fixed blurring kernel and additive noise, and tested our codes on a series of  $\beta$  values. In this experiment, Gaussian kernel was applied with blurring size  $\text{hsize} = 11$ , standard deviation  $\sigma = 10$  and noise level  $\text{std}=10^{-3}$ . We terminated Algorithm 1 with two different tolerance, i.e.,  $\epsilon = 10^{-3}$  and  $10^{-4}$  in (2.11). The SNR and CPU time results varying as functions of  $\beta$  are given in Figure 3.1.

It can be seen from Figure 3.1 that, for any fixed  $\beta$ , the results from  $\epsilon = 10^{-4}$  (bottom left) do no substantially better than those from  $\epsilon = 10^{-3}$  (top left) in the sense of higher SNR. On the other hand, from the two plots on the right-hand side in Figure 3.1, higher accuracy requires longer CPU time. In practice, spending much longer time to attain little improvement in recovery quality is more a loss than gain. Moreover, the observation  $f$  is more often than not noisy, and further experiments demonstrated that the tolerance value should be consistent with the noise level. Therefore, in all experiments we set  $\epsilon$  to be consistent with the noise level, instead of solving TV deconvolution problems to an extremely high accuracy. It is also clear from Figure 3.1 that the quality of recovered images has little relevance to the value of  $\beta$ . As

<sup>1</sup>ADM for TV deconvolution is implemented as version 4.0 of the FTVd package which is held at Rice University and available at <http://www.caam.rice.edu/~optimization/L1/ftvd/v4.0/>.

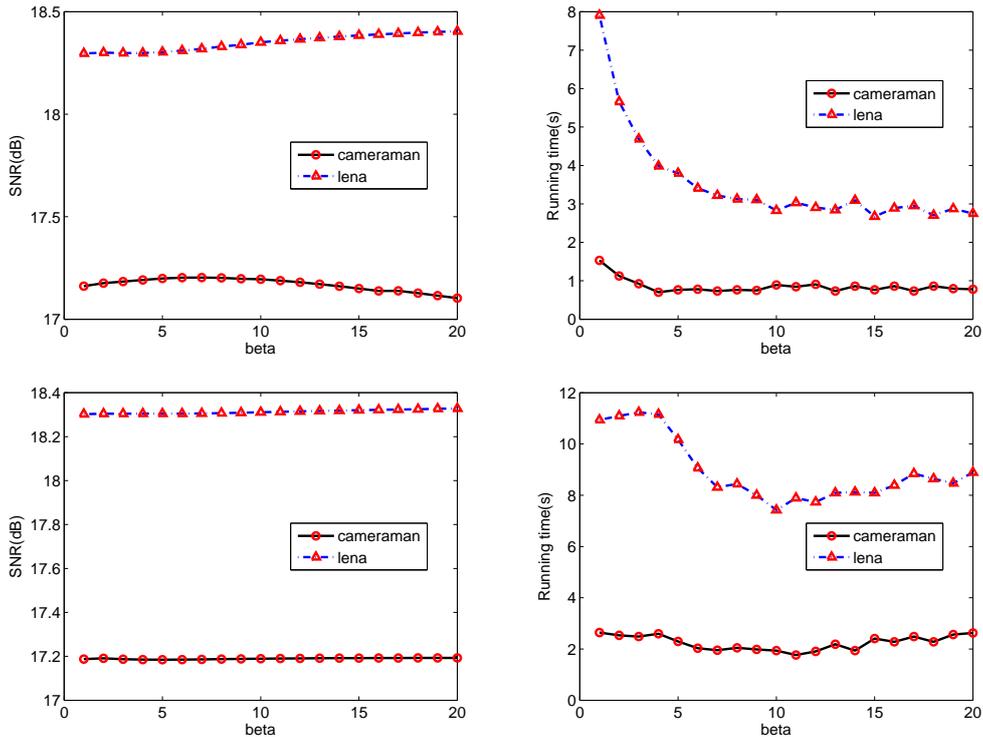


FIG. 3.1. SNR (left) and running time (right) results of ADM corresponding to a set of  $\beta$  values. First row: terminated by (2.11) with  $\epsilon = 10^{-3}$ ; Second row: terminated by (2.11) with  $\epsilon = 10^{-4}$ .

a matter of fact, Algorithm 1 converges to (1.4) for any fixed  $\beta > 0$  in theory. In comparison, the speed of convergence is more or less related to  $\beta$ . Based on our experimental results, a  $\beta$  value around 10 performs favorably. Therefore, we set  $\beta = 10$  in all tests. When comparing ADM with FTVd, we initialized both algorithms at the blurry and noisy image, and all parameters in FTVd were set to be default.

**3.2. Comparison results with FTVd.** In this section, we present two classes of results to compare ADM with FTVd on solving TV/ $L^2$  problem (1.4). In the first class of results, we tested two types of blurring kernels from MATLAB, i.e., Gaussian and average. For each type of kernel, we compared the two algorithms on a set of kernel sizes. The additive noise used was Gaussian with mean zero and standard deviation  $10^{-3}$ . This level of noise is substantial in the context of deblurring. ADM was terminated using (2.11) with  $\epsilon = 10^{-3}$  and FTVd was stopped with default setting. Detailed information on the setup of different tests is summarized in Table 3.2. The SNR and CPU time results are given in Figure 3.2.

TABLE 3.1  
Information on setup for four tests.

Test No.	Image	Size	Blurring type	Blurring kernel parameters
1	Cameraman	$256 \times 256$	Gaussian	$\mathbf{hsize} = \{3, 5, \dots, 15\}, \sigma = 10$
2	Lena	$512 \times 512$	Gaussian	$\mathbf{hsize} = \{3, 5, \dots, 15\}, \sigma = 10$
3	Cameraman	$256 \times 256$	Average	$\mathbf{hsize} = \{3, 5, \dots, 15\}$
4	Lena	$512 \times 512$	Average	$\mathbf{hsize} = \{3, 5, \dots, 15\}$

It can be seen from Figure 3.2 that both algorithms consumed only a few seconds to recover an image

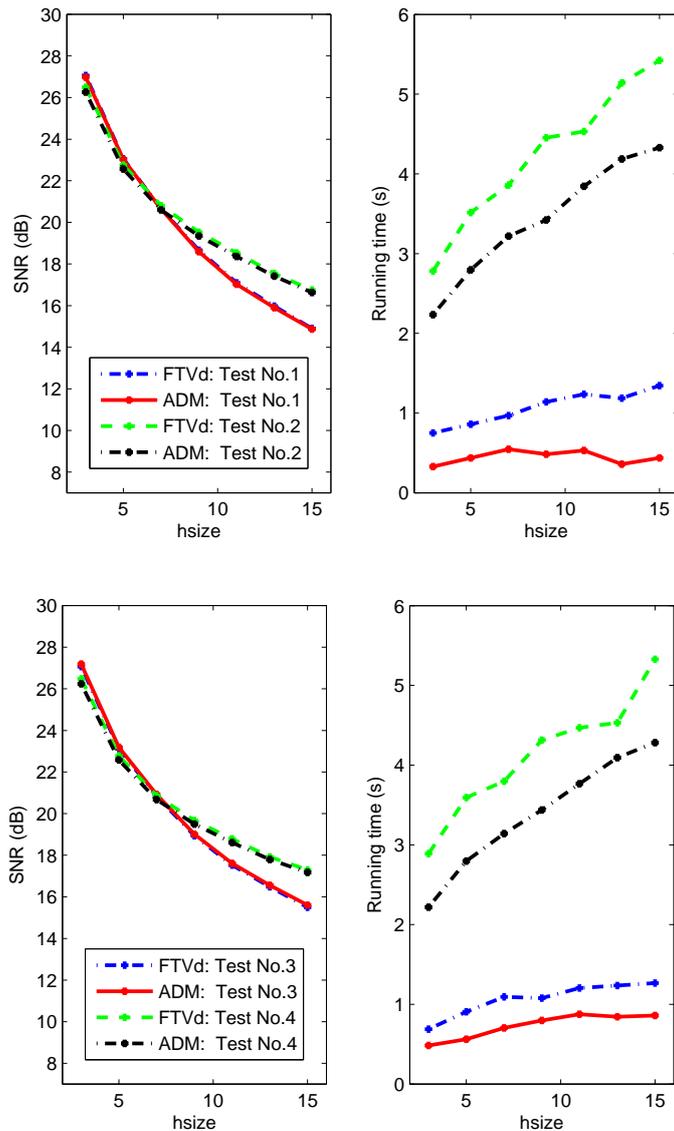


FIG. 3.2. SNR and running time results on tests No. 1, 2, 3 and 4.

of size as large as  $512 \times 512$ . The increase in CPU time for both algorithms when the underlying image becomes larger is acceptable, which is because the main computation of both methods is several FFTs. Since both algorithms solve (1.4), the resulting SNR values from FTVd and ADM are approximately identical for each  $\mathbf{hsize}$ . However, ADM is faster than FTVd, which is mainly because the augmented Lagrangian based alternating minimization is far better than the penalty method with continuation as implemented in FTVd. We note that the the faster convergence of ADM compared to FTVd becomes more obvious for TV/L<sup>1</sup> problems and larger sized images, see comparison results presented in Figures 4.2, 4.3 and 4.4.

In the second class of results, we compared ADM with FTVd under different level of Gaussian noise. In this experiment, we blurred the Lena image with the average blur of  $\mathbf{hsize} = 13$ . We ran both algorithms twice corresponding to two levels of Gaussian noise with mean zero and different standard deviations. Specifically, we tested  $\mathbf{std} = 10^{-2}$  and  $10^{-3}$ , and ADM was terminated by (2.11) with  $\epsilon = 10^{-2}$  and  $10^{-3}$ ,



FIG. 3.3. Recovered results from FTVd and ADM. In both rows from left to right: blurry and noisy image (B&N), results recovered by FTVd and ADM, respectively. CPU and It. represent running time and iteration numbers, respectively. Top row:  $\text{std}=10^{-2}$ ; Bottom row:  $\text{std}=10^{-3}$ .

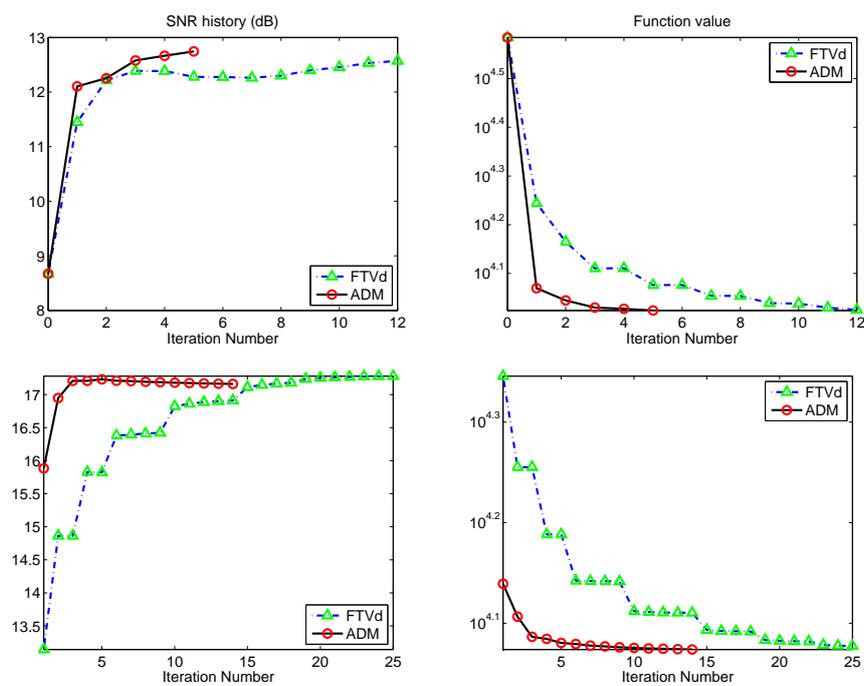


FIG. 3.4. SNR (left) and function values (right) results of FTVd and ADM. In all plots, x-axes represent iteration numbers. Top row:  $\text{std}=10^{-2}$ ; Bottom row:  $\text{std}=10^{-3}$ .

respectively. FTVd was stopped with default setting. The blurry and noisy images (B&N) and the restored ones are presented in Figure 3.3. The changing of SNR and objective values as functions of iteration numbers are also presented in Figure 3.4.

It can be seen from Figure 3.3 that ADM obtained images of comparable quality as those from FTVd in much less CPU time. As mentioned before, the per-iteration computation of both ADM and FTVd for grayscale image deconvolution is mainly two FFTs, each at a cost of  $O(n^2 \log(n))$ . Therefore, the running time consumed by the two algorithms is proportional to the iteration numbers. From Figure 3.4 ADM is able to reach a reasonable SNR in merely several iterations and decrease the objective function rapidly. To reach a solution of similar quality, FTVd generally takes more iterations. For example, for  $\text{std}=10^{-2}$ , the highest SNR value 12.74dB is reached by ADM in 5 iterations, while the result of FTVd is not better after 12 iteration, see the top left plot in Figure 3.4. Similarly, for  $\text{std}=10^{-3}$  (bottom plots in Figure 3.4), the highest SNR value 17.33dB is reached in merely 5 iterations, while FTVd took at least 19 iterations to obtain a comparable value. Since FTVd implements the classical penalty method with continuation on  $\beta$ , these results validate that the TV/L<sup>2</sup> model (1.4) is not well approximated when  $\beta$  is not sufficiently large. In pure optimization point of view, ADM reached smaller function values than FTVd throughout the whole iteration process, see plots on the right-hand side of Figure 3.4 (the objective and SNR values at the initial point were omitted so that the function values belong to a small dynamic range). We note that, for  $\text{std}=10^{-3}$ , the SNR value started to decrease a little bit after ADM obtained the highest value 17.23dB, which is because, in the presence of noise, the true solution of (1.4) does not necessarily have higher SNR than mediate iterates, which can be validated by examining the function values from ADM (plot on bottom right of Figure 3.4) because they kept decreasing.

From the above comparison results, it is safe to conclude that ADM is more efficient than FTVd when applied to (1.4). In the presence of noise, as is always the case in practical applications, the solution of (1.4) is merely an approximation to the true image. Therefore, solving (1.4) to an extremely high accuracy is not always beneficial and thus is not necessarily. In real-time imaging applications where speed is crucial, a numerical algorithm which converges fast at the beginning iterations, like ADM, is highly useful.

**4. Extensions.** In this section, we extend the ADM approach to solve the multichannel variants of (1.4) and (1.5). The derivation of algorithms is similar to that of Algorithm 1. We also present experimental results and compare with FTVd to show the efficiency of ADM.

**4.1. Extension to multichannel variant of TV/L<sup>2</sup>.** Suppose  $\bar{x} = [\bar{x}^{(1)}; \dots; \bar{x}^{(m)}] \in \mathbb{R}^{mn^2}$  is an  $m$ -channel image of size  $n \times n$ , where  $\bar{x}^{(j)}$  represents the  $j$ th channel,  $j = 1, \dots, m$ , and its blurry and noisy observation  $f = [f^{(1)}; \dots; f^{(m)}] \in \mathbb{R}^{mn^2}$  is given by (1.1) in which case  $K = [K_{k,l}]_{k,l=1}^m \in \mathbb{R}^{mn^2 \times mn^2}$  is a cross-channel blurring operator. We recover  $\bar{x}$  via solving the following local weighted TV/L<sup>2</sup>-like problem

$$\min_x \sum_i \|(I_m \otimes D_i)x\| + \frac{\mu}{2} \|Kx - f\|^2, \quad (4.1)$$

where  $I_m$  is the identity matrix of order  $m$ , “ $\otimes$ ” represents the Kronecker product, and  $I_m \otimes D_i$  is a local finite difference operator at pixel  $i$ . For more details on this multichannel TV deconvolution model, see e.g., [37]. In the following, we show that ADM can be applied to solve (4.1).

For simplicity, we let  $G_i = I_m \otimes D_i$ ,  $i = 1, \dots, n^2$ . Simple manipulation shows that (4.1) is equivalent to

$$\min_{y,x} \left\{ \sum_i \|y_i\| + \frac{\mu}{2} \|Kx - f\|^2 : y_i = G_i x, i = 1, \dots, n^2 \right\}. \quad (4.2)$$

For  $j = 1, \dots, 2m$ , let  $G^{(j)} \in \mathbb{R}^{n^2 \times mn^2}$  be the matrix formed by staking the  $j$ th rows of  $G_1, G_2, \dots, G_{n^2}$ .

Denote

$$G \triangleq \begin{pmatrix} G^{(1)} \\ \vdots \\ G^{(2m)} \end{pmatrix} \in \mathbb{R}^{2mn^2 \times mn^2} \text{ and } \mathbf{Y} \triangleq \begin{pmatrix} \mathbf{y}_1^\top \\ \vdots \\ \mathbf{y}_{n^2}^\top \end{pmatrix} \triangleq [y_1, y_2, \dots, y_{2m}] \in \mathbb{R}^{n^2 \times 2m}. \quad (4.3)$$

Furthermore, we let  $y = [y_1; \dots; y_{2m}]$ . Given this notation, the ADM applied to (4.2) can be sketched as follows. First, we form the augmented Lagrangian function  $\mathcal{L}_{\mathcal{A}}(x, y, \lambda)$  of (4.1), where  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_{2m}] \in \mathbb{R}^{2mn^2}$  is the Lagrangian multiplier. Then we apply alternating minimization to  $\mathcal{L}_{\mathcal{A}}(x, y, \lambda)$  with respect to  $x$  and  $y$  and immediate updates of multipliers. Started at  $(x^k, \lambda^k)$ , the ADM applied to (4.1) obtains  $(x^{k+1}, \lambda^{k+1})$  in three steps given below.

1. Compute  $y^{k+1}$  via

$$\mathbf{y}_i^{k+1} = \max \left\{ \|G_i x^k + \lambda_i^k / \beta\| - \frac{1}{\beta}, 0 \right\} \frac{G_i x^k + \lambda_i^k / \beta}{\|G_i x^k + \lambda_i^k / \beta\|}, \quad i = 1, \dots, n^2. \quad (4.4)$$

2. Compute  $x^{k+1}$  via solving

$$\left( G^\top G + \frac{\mu}{\beta} K^\top K \right) x = G^\top (y^{k+1} - \lambda^k / \beta) + \frac{\mu}{\beta} K^\top f. \quad (4.5)$$

3. Update  $\lambda$  via

$$\lambda^{k+1} = \lambda^k - \gamma \beta (y^{k+1} - G x^{k+1}). \quad (4.6)$$

Recall that  $D = (D^{(1)}, D^{(2)})$  is the global finite difference operator. For images in the standard RGB (red, green and blue) color system, i.e.,  $x = [x^r; x^g; x^b] \in \mathbb{R}^{3n^2}$ , equations (4.5) reduce to

$$\left( I_3 \otimes (D^\top D) + \frac{\mu}{\beta} K^\top K \right) x = (I_3 \otimes D)^\top (y^{k+1} - \lambda^k / \beta) + \frac{\mu}{\beta} K^\top f. \quad (4.7)$$

Under the periodic boundary conditions for  $x$ , the coefficient matrix in (4.7) has a block circulant structures and can be solved by 6 FFTs (including 3 inverse FFTs) and a block-wise Gaussian elimination. For implementation details on (4.4)-(4.6), see [37] for similar discussions.

To compare the ADM framework (4.4)-(4.6) with FTVd for multichannel image restoration, we ran two tests with within- and cross- channel blurs, respectively. Let the cross-channel kernel  $H$  be of the form

$$H = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix}$$

where  $H_{ii}$ ,  $i = 1, 2, 3$ , are within channel kernels and  $H_{ij}$ ,  $i \neq j$ , are cross-channel kernels. For simplicity, let the motion blur with a motion length `len` and an angle `theta` in the counterclockwise direction be denoted by `M(len, theta)`. Similarly, the Gaussian blur with a square support size `hsize` and a standard deviation `sigma` is denoted by `G(hsize, sigma)`, and the average blur with a square support size `hsize` by `A(hsize)`. In our experiments, we generated cross-channel blurring kernels in the same manner as in [37] which is described below.

1. Generate 9 kernels:

$$\{\text{M}(11, 45), \text{M}(21, 90), \text{M}(41, 135), \text{G}(7, 5), \text{G}(9, 5), \text{G}(11, 5), \text{A}(13), \text{A}(15), \text{A}(17)\};$$

2. Randomly assign the above 9 kernels to  $\{H_{11}, H_{12}, H_{13}, H_{21}, H_{22}, H_{23}, H_{31}, H_{32}, H_{33}\}$ ;
3. Multiply each  $H_{ij}$  with  $w_{ij}$ , where  $W \in \mathbb{R}^{3 \times 3}$  and  $w_{ij}$  is the  $(i, j)$ th element of  $W$ .

In the first test, we used within-channel blurring, i.e., setting  $W$  to be the identity matrix of order 3. In the second test, we tested cross-channel kernels with

$$W = \begin{pmatrix} 0.7 & 0.15 & 0.15 \\ 0.1 & 0.8 & 0.1 \\ 0.2 & 0.2 & 0.6 \end{pmatrix}.$$

In both tests, we added Gaussian noise with zero mean and standard deviation  $\text{std}=10^{-3}$ . All parameters were set to be identical to those described in Section 3, except we terminated ADM by (2.11) with  $\epsilon = 5 \times 10^{-3}$ . The blurry and noisy images and recovered ones by both methods are presented in Figure 4.1. Similar to the results for grayscale images, ADM generally consumed less CPU time than FTVd to obtain an image of comparable quality. We also studied the convergence behavior of SNR and objective function values as functions of iteration numbers. The results are generally similar to those presented in Figure 3.4 and thus are omitted.



FIG. 4.1. RGB image results from FTVd and ADM. In both rows from left to right: blurry and noisy image (B&N), results recovered by FTVd and ADM, respectively. Top row: results recovered from within channel blurring; Bottom row: results recovered from cross-channel blurring.

**4.2. Extensions to TV/L<sup>1</sup> and its multichannel variant.** In this section, we further extend the ADM approach to solve (1.5) and its multichannel variant, which are suitable for impulsive noise removal. We merely treat the single-channel case and the treatment of multichannel variant is completely analogous. Similar as previous discussions, by introducing variables (1.5) is equivalently transformed to

$$\min_{x,y,z} \left\{ \sum_i \|y_i\| + \mu \|z\|_1 : y_i = D_i x, i = 1, \dots, n^2, z = Kx - f \right\}. \quad (4.8)$$

The augmented Lagrangian function of (4.8) is

$$\begin{aligned} \mathcal{L}_{\mathcal{A}}(x, y, z, \lambda) &= \sum_i \|\mathbf{y}_i\| - \lambda_1^\top (y - Dx) + \frac{\beta_1}{2} \sum_i \|\mathbf{y}_i - D_i x\|^2 \\ &\quad + \mu \|z\|_1 - \lambda_2^\top [z - (Kx - f)] + \frac{\beta_2}{2} \|z - (Kx - f)\|^2, \end{aligned} \quad (4.9)$$

where  $\beta_1, \beta_2 > 0$ , and  $\lambda = (\lambda_1, \lambda_2) \in \mathbb{R}^{3n^2}$  is the Lagrangian multiplier. According to the scheme of ADM, for given  $(x^k, \lambda^k)$ , the next iterate  $(x^{k+1}, y^{k+1}, z^{k+1}, \lambda^{k+1})$  is generated as follows.

1. Fixing  $x = x^k$  and  $\lambda = \lambda^k$  and minimizing  $\mathcal{L}_{\mathcal{A}}$  with respect to  $y$  and  $z$  to obtain  $y^{k+1}$  and  $z^{k+1}$ .

The minimizers are given explicitly by

$$\mathbf{y}_i^{k+1} = \max \left\{ \left\| D_i x^k + \frac{(\lambda_1)_i^k}{\beta_1} \right\| - \frac{1}{\beta_1}, 0 \right\} \frac{D_i x^k + (\lambda_1)_i^k / \beta_1}{\|D_i x^k + (\lambda_1)_i^k / \beta_1\|}, \quad i = 1, 2, \dots, n^2, \quad (4.10)$$

$$z^{k+1} = \text{sgn}(Kx^k - f + (\lambda_2)^k / \beta_2) \circ \max\{|Kx^k - f + (\lambda_2)^k / \beta_2| - \mu / \beta_2, 0\}, \quad (4.11)$$

where  $|\cdot|$  and  $\text{sgn}$  represent componentwise absolute value and signum function, respectively.

2. Compute  $x^{k+1}$  via solving the normal equations

$$\left( D^\top D + \frac{\beta_2}{\beta_1} K^\top K \right) x = D^\top \left( y^{k+1} - \frac{(\lambda_1)^k}{\beta_1} \right) + K^\top \left( \frac{\beta_2 z^{k+1} - (\lambda_2)^k}{\beta_1} \right) + \frac{\beta_2}{\beta_1} K^\top f. \quad (4.12)$$

3. Update  $\lambda$  via

$$(\lambda_1)^{k+1} = (\lambda_1)^k - \gamma \beta_1 (y^{k+1} - Dx^{k+1}), \quad (4.13)$$

$$(\lambda_2)^{k+1} = (\lambda_2)^k - \gamma \beta_2 [z^{k+1} - (Kx^{k+1} - f)]. \quad (4.14)$$

Now we present experimental results to compare ADM with FTVd on solving TV/L<sup>1</sup> problems. In addition to the grayscale image cameraman, we also tested a RGB image Rose (303×250). In our experiments, we first generated a blurry image and then corrupted 60% of its pixels by salt-and-pepper noise. In our experiments, we simply set  $\beta_1 = 5$  and  $\beta_2 = 20$ , which may not be optimal but perform quite well in our experiments. For 60% impulsive noise corruption, we set  $\mu = 4$  in (1.5) as suggested in [34]. We terminated ADM by (2.11) with  $\epsilon = 10^{-3}$ . The blurring kernels and detailed recovery results are given in Figures 4.2 and 4.3. For the experiment on Rose image, we also recorded the SNR and objective values to better compare the performance of FTVd and ADM. The changing behavior of function values and SNR with respect to iteration numbers is presented in Figure 4.4.

It can be seen from Figures 4.2 and 4.3 that, to obtain comparable or equally good images, ADM consumes much less iterations than FTVd and thus is much faster. Compared with FTVd, the faster convergence of ADM becomes even more obvious on TV/L<sup>1</sup> than on TV/L<sup>2</sup> problems. From Figure 4.2, ADM took only 2.6 seconds to recover the cameraman image, while FTVd consumed 21.9 seconds to obtain an equally good image. For color image, the efficiency of ADM becomes more obvious, see the test results on Rose image given in Figure 4.3, which is because FTVd needs one more penalty term for TV/L<sup>1</sup> problems, while ADM converges for small values of  $\beta$ 's and does not need continuation. From Figure 4.4, it can be seen that jumps appear in the SNR and function value curves of FTVd, which is due to the switching of subproblems ( $\beta_1$  and  $\beta_2$  increase in a continual scheme), while the results of ADM changes smoothly.



FIG. 4.2. Grayscale image results of FTVd and ADM on  $TV/L^1$ . Left: blurry and noisy image (blurred by average kernel with  $hsize=9$  and corrupted by 60% salt-and-pepper noise); Middle: recovered by FTVd; Right: recovered by ADM.



FIG. 4.3. RGB image results of FTVd and ADM on  $TV/L^1$  problem. Left: blurry and noisy image (blurred by the same cross-channel kernel as used in Section 4.1 and corrupted by 60% salt-and-pepper noise); Middle: recovered by FTVd; Right: recovered by ADM.

**5. Concluding remarks.** Based on the classical augmented Lagrangian function and an alternating minimization framework, we propose the use of alternating direction method for recovering images from blurry and noisy observations. The ADM approach is applicable to  $TV/L^2$ ,  $TV/L^1$  and their multichannel extensions. Extensive comparison results on single- and multi-channel images with different types of blurs and noise indicate that ADM is stable, efficient and in particular faster than a state-of-the-art algorithm FTVd [38]. And its excellent performance depends on fewer fine tuning of parameters than FTVd. The same as FTVd, ADM also takes advantage of the convolution structure. The per-iteration cost of ADM is dominated by shrinkage operations and FFTs. For  $TV/L^1$  problems, the faster convergence of ADM becomes more obvious, which is mainly due to the advantage of iterative updates of multipliers instead of increasing penalty parameters to a large value.

#### REFERENCES

- [1] L. Ambrosio and V. Tortorelli, *Approximation of functionals depending on jumps by elliptic functionals via  $\Gamma$ -convergence*, Comm. Pure Appl. Math., 43, pp. 999–1036, 1990.
- [2] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2, pp. 183–202, 2009.
- [3] J. M. Bioucas-Dias and M. A. T. Figueiredo, *A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration*, IEEE Trans. Image Process., 16, pp. 2992–3004, 2007.
- [4] C. H. Chen, B. S. He and X. M. Yuan, *Matrix completion via alternating direction method*, submitted, 2009.
- [5] D. Dobson, and F. Santosa, *Recovery of blocky images from noisy and blurred data*, SIAM J. Appl. Math., 56, pp.

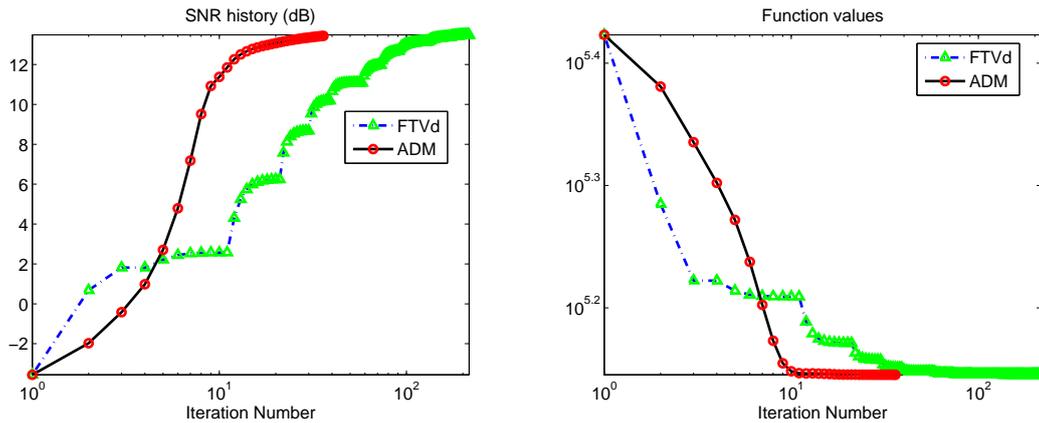


FIG. 4.4. Comparison of FTVd and ADM on the test of Rose image. Left: SNR results; Right: function value results.

1181–1198, 1996.

- [6] J. Douglas and H. H. Rachford, *On the numerical solution of heat conduction problems in two and three space variables*, Transactions of the American mathematical Society, 82, pp. 421–439, 1956.
- [7] J. Eckstein, and D. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Progr., 55, North-Holland, 1992.
- [8] M. Elad, *Why simple shrinkage is still relevant for redundant representations?*, IEEE Transactions on Information Theory, 52 (2006), pp. 5559–5569.
- [9] M. Elad, B. Matalon, and M. Zibulevsky, *Image denoising with shrinkage and redundant representations*, in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, 2006.
- [10] E. Esser, *Applications of Lagrangian-Based alternating direction methods and connections to split Bregman*, UCLA CAM Report 09–31, 2009.
- [11] M. Fukushima, *Application of the alternating direction method of multipliers to separable convex programming*, Comput. Optim. Appl., 1 (1992), pp. 93–111.
- [12] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appl., 2 (1976), 17–40.
- [13] R. Glowinski, and A. Marrocco, *Sur l'approximation par elements finis d'ordre un, et la resolution par penalisation-dualite d'une classe de problemes de Dirichlet nonlineaires*, Rev. Francaise d'Aut. Inf. Rech. Oper., 2 (1975), pp. 41–76.
- [14] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.
- [15] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*, SIAM Studies in Applied Mathematics, Philadelphia, PA, 1989.
- [16] T. Goldstein and S. Osher, *The split Bregman method for L<sub>1</sub>-Regularized Problems*, SIAM J. Imag. Sci., 2 (2009), pp. 323–343.
- [17] R. Gonzalez and R. Woods, *Digital image Processing*, Addison-Wesley, Reading, MA, 1992.
- [18] B. S. He, H. Yang, *Some Convergence Properties of a Method of multipliers for linearly constrained monotone variational inequalities*, Operations Research Letters, 23 (1998), pp. 151–161.
- [19] B. S. He, M. H. Xu and X. M. Yuan, *Solving large-scale least squares covariance matrix problems by alternating direction methods*, submitted, 2009.
- [20] M. R. Hestenes, *Multiplier and gradient methods*, J. Optimiz. Theory App., vol. 4, pp. 303–320, 1969.
- [21] D. Mumford and J. Shah, *Optimal approximations by piecewise smooth functions and associated variational problems*, Commun. Pure Appl. Math., 42 (1989), pp. 577–685.
- [22] M. K. Ng, R. H. Chan, and W.-C. Tang, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–866.
- [23] M. Nikolova, *Local strong homogeneity of a regularized estimator*, SIAM J. Appl. Math., 61 (2000), pp. 633–658.
- [24] L. I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, 60 (1992), pp. 259–268.
- [25] L. Rudin and S. Osher, *Total variation based image restoration with free local constraints*, Proc. 1st IEEE ICIP, 1 (1994), pp. 31–35.

- [26] J. Shah, *A common framework for curve evolution, segmentation and anisotropic diffusion*, in Proc. IEEE Conf. CVPR, (1996), pp. 136–142.
- [27] J. L. Starck, M. Nguyen, and F. Murtagh, *Wavelets and curvelets for image deconvolution: a combined approach*, Signal Processing, 83 (2003), pp. 2279–2283.
- [28] A. Tikhonov, and V. Arsenin, *Solution of Ill-Posed problems*, Winston, Washington, DC, 1977.
- [29] P. Tseng, *Applications of Splitting Algorithm to Decomposition in Convex Programming and Variational Inequalities*, SIAM Journal on Control and Optimization, 29 (1991), pp. 119–138.
- [30] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative reularization method for total variational-based image restoration*, SIAM Multiscale Model. and Simu, 4 (2005), pp. 460–489.
- [31] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, New York, NY, pp. 283–298, 1969.
- [32] J. L. Starck, E. Candès, and D. Donoho, *Astronomical image representation by the curvelet transform*, Astronomy and Astrophysics, 398 (2003), pp. 785–800.
- [33] C. R. Vogel and M. E. Oman, *Iterative methods for total variation denoising*. SIAM J. Sci. Comput., 17 (1996), pp.227–238.
- [34] Y. Wang, J. Yang, W. Yin, and Y. Zhang, *A new alteranting minimization alrorithm for total variation image reconstruction*, SIAM J. Imaging Sci., 1 (2008), pp. 948–951.
- [35] Z. Wen, D. Goldfarb and W. Yin, *Alternating direction augmented Lagrangian methods for semidefinite programming*, submitted for publication (2009).
- [36] —, *A fast, robust total variation based reconstruction of noisy, blurred images*, IEEE Trans. Image Process., 7 (1998), pp. 813–824.
- [37] J. Yang, W. Yin, Y. Zhang, and Y. Wang, *A fast algorithm for edge-preserving variational multichannel image Rstoration*, SIAM J. Imaging Sci., 2 (2009), pp. 569–592.
- [38] J. Yang, Y. Zhang and W. Yin, *An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise*, SIAM J. Sci. Comput., 31 (2009), pp. 2842–2865.
- [39] J. Yang, Y. Zhang, and W. Yang, *A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data*, J. Selected Topics in Signal Processing, special issue on Compressive Sensing, submitted, 2008.
- [40] C. H. Ye, and X. M. Yuan, *A Descent Method for Structured Monotone Variational Inequalities*, Optimization Methods and Software, 22 (2007), pp. 329–338.
- [41] X. Yuan, and J. Yang, *Sparse and low-rank matrix decomposition via alternating direction methods*, submitted, 2009.
- [42] Y. Zhang, *Your ALgorithm for L1*, Available at: <http://www.caam.rice.edu/~yzhang/YALL1/>