SIAM J. IMAGING SCIENCES
Vol. 6, No. 1, pp. 680–697

Constrained Total Variation Deblurring Models and Fast Algorithms Based on Alternating Direction Method of Multipliers*

Raymond H. Chan†, Min Tao‡, and Xiaoming Yuan§

Abstract. The total variation (TV) model is attractive in that it is able to preserve sharp attributes in images. However, the restored images from TV-based methods do not usually stay in a given dynamic range, and hence projection is required to bring them back into the dynamic range for visual presentation or for storage in digital media. This will affect the accuracy of the restoration as the projected image will no longer be the minimizer of the given TV model. In this paper, we show that one can get much more accurate solutions by imposing box constraints on the TV models and solving the resulting constrained models. Our numerical results show that for some images where there are many pixels with values lying on the boundary of the dynamic range, the gain can be as great as 10.28 decibel in the peak signal-to-noise ratio. One traditional hindrance using the constrained model is that it is difficult to solve. However, in this paper, we propose using the alternating direction method of multipliers (ADMM) to solve the constrained models. This leads to a fast and convergent algorithm that is applicable for both Gaussian and impulse noise. Numerical results show that our ADMM algorithm is better than some state-of-the-art algorithms for unconstrained models in terms of both accuracy and robustness with respect to the regularization parameter.

Key words. total variation, deblurring, alternating direction method of multipliers, box constraint

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

DOI. 10.1137/110860185

1. Introduction.

In this paper, we consider the problem of deblurring digital images under Gaussian or impulse noise. Without loss of generality, we consider all images as square images of size n-by-n. Let $\bar{x} \in \mathbb{R}^{n^2}$ be a given original image concatenated into an n^2-vector, let $K \in \mathbb{R}^{n^2 \times n^2}$ be a blurring operator acting on the image, and let $\omega \in \mathbb{R}^{n^2}$ be the Gaussian or impulse noise added onto the image. The observed image $f \in \mathbb{R}^{n^2}$ can be modeled by $f = K\bar{x} + \omega$, and our objective is to recover $\bar{x}$ from f.

It is well known that recovering $\bar{x}$ from f by directly inverting K is unstable and can produce a very noisy result because K is highly ill-conditioned. Instead one usually solves

$$\min_x \{\Phi_{\mathrm{reg}}(x) + \mu \Phi_{\mathrm{fit}}(x, f)\}, \tag{1.1}$$

*Received by the editors December 22, 2011; accepted for publication (in revised form) December 26, 2012; published electronically March 28, 2013.
http://www.siam.org/journals/siims/6-1/86018.html

†Department of Mathematics, The Chinese University of Hong Kong, Shatin, NT, Hong Kong, People's Republic of China (rchan@math.cuhk.edu.hk). This author's work was supported in part by HKRGC grant CUHK400510 and CUHK Direct Allocation grant 2060408.

‡School of Science, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210093, People's Republic of China (taom@njupt.edu.cn). This author's work was supported by the Scientific Research Foundation of Nanjing University of Posts and Telecommunications (NY210049).

§Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong, People's Republic of China (xmyuan@hkbu.edu.hk). This author's work was supported by the Hong Kong General Research Fund: 203311.

where $\Phi_{\rm reg}(x)$ regularizes the solution by enforcing certain prior constraints, $\Phi_{\rm fit}(x,f)$ measures how fit x is to the observation f, and μ is the regularization parameter balancing these two terms. Traditional choices for $\Phi_{\rm reg}(x)$ include the Tikhonov-like regularization [36], the total variation (TV) regularization [29], and the Mumford–Shah regularization [23] and its variants [1, 31]. In this paper, we consider the TV regularization [29, 28] as it has been shown to preserve sharp edges both experimentally and theoretically. For $\Phi_{\rm fit}(x)$, we consider $\|Kx-f\|_2^2$ and $\|Kx-f\|_1$, which are, respectively, suitable data-fitting terms for images corrupted by Gaussian [38, 41, 42] and impulse noise [2, 7, 9, 40, 41]. The corresponding problems (1.1) with TV regularization are called the TV-L2 and TV-L1 problems, respectively. There are many good existing algorithms for solving these problems, for example, [29, 28, 39, 38, 41, 42, 27, 9, 2, 7, 40] to mention just a few. In the literature, some authors also discuss other nonquadratic fidelity terms besides the L1 fidelity term; see, e.g., [13, 32, 33].

In this paper, we consider the case where the images are digital images so that their pixel values have to lie in a certain dynamic range $[l,u]$. For example, for 8-bit images, we have $[l,u]=[0,255]$. Notice that for many existing algorithms such as those listed in the previous paragraph, their restored image x will not necessarily be in $[l,u]$. Therefore if x is to be stored or displayed digitally, its pixel values must first be projected onto $[l,u]$. There are many ways to do this. One can just map all pixels with values that are less than l to l and those that are greater than u to u. We call this *truncation*. Another way is to linearly stretch the pixel values of x to $[l,u]$ by a linear mapping, which we call *stretching*. The MATLAB command `imshow` provides both kinds of projections, with *stretching* being the default method. Clearly, after projection, the image no longer minimizes the unconstrained model. In fact, we will see in the numerical examples in section 4 that this minimize-and-project approach usually gives inferior solutions.

For digital image restoration, a more accurate model for x is to explicitly constrain the solution in $[l,u]$; i.e., we solve the constrained model:

$$\min_{x\in\Omega}\{\Phi_{\rm reg}(x)+\mu\Phi_{\rm fit}(x,f)\}. \tag{1.2}$$

Here $\Omega=\{x\in\mathbb{R}^{n^2}\mid l\le x\le u\}$ with $l,u\in\mathbb{R}_+^{n^2}$, and the constraints are to be interpreted entrywise; i.e., $l_i\le x_i\le u_i$ for any $1\le i\le n^2$. Constrained TV-L2 models have recently been considered in [4], where their numerical tests indicate that one can get more than 2 decibel (dB) improvement in the peak signal-to-noise ratio (PSNR) for some special images simply by imposing the box constraint in the TV-L2 model. (See (4.1) for the definition of PSNR.) Our numerical experiments in section 4 reveal that the improvement can even be as big as 10.28dB for an image with all pixel values either at l or at u. It is therefore advantageous to solve the constrained model (1.2) directly instead of using the minimize-and-project approach, provided that we have an efficient solver at our disposal.

Constrained problems are usually much more difficult to solve than the unconstrained ones. However, there are some existing methods that solve the constrained image restoration model (1.2). For constrained L2-L2 problems, i.e., where the regularization term is the L2-norm of some derivative of x, there are several methods that are based on Newton-like methods; see [10, 22]. For constrained TV problems, the singularity of the TV functional prohibits the application of Newton-like methods. Recently, Beck and Teboulle [4] proposed a fast gradient-based algorithm for solving constrained TV-L2 problems. As far as we know, there are currently no solvers for constrained TV-L1 problems. In this paper, we derive a solver for both the constrained TV-L2 and TV-L1 problems. Our solver is based on the alternating direction method of multipliers (ADMM) which was developed in the 1970s [18, 17]. The convergence of our algorithms is thus guaranteed by the classical theory in ADMM literature, e.g., [21]. We compare our algorithms with the state-of-the-art solvers, such as fast TV deblurring (FTVd) [38] and the augmented Lagrange method (ALM) [40] for the unconstrained model (1.1), and the monotone fast iterative shrinkage/threshholding algorithm (MFISTA) [4] for the constrained model (1.2). Numerical results show that our algorithms are faster than MFISTA for solving the same model while yielding more accurate restored images than those from the unconstrained model. Also, our algorithms are more robust with respect to the changes in the regularization parameter μ.

The rest of this paper is organized as follows. In section 2, we recall briefly existing solvers for unconstrained TV-L1 and TV-L2 problems. In section 3, we derive our ADMM-based algorithms to solve the constrained TV-L1 and TV-L2 problems. In section 4, numerical comparisons with existing methods are carried out to confirm the effectiveness of our approach. Finally, some concluding remarks are made in section 5.

2. TV deblurring models and solvers.

In this section, we briefly review some related methods for solving TV deblurring problems. We start with the TV-L2 model, which is good for deblurring images under the corruption by Gaussian noise [29, 28, 38, 41, 42, 27]:

$$\min_x \left\{ \sum_{i=1}^{n^2} \|D_i x\|_2 + \frac{\mu}{2}\|Kx - f\|_2^2 \right\}, \tag{2.1}$$

where $D_i x \in \mathbb{R}^2$ represents the first-order finite difference of x at pixel i in both horizontal and vertical directions. More specifically, let $x = (x_1, x_2, \ldots, x_{n^2})^\top$, and let x be extended by a periodic boundary condition. Then $D_i x := ((D^{(1)}x)_i, (D^{(2)}x)_i)^\top \in \mathbb{R}^2$ $(i = 1, \ldots, n^2)$, where

$$(D^{(1)}x)_i := \begin{cases} x_{i+n} - x_i & \text{if } 1 \le i \le n(n-1), \\ x_{\text{mod}(i,n)} - x_i & \text{otherwise,} \end{cases} \tag{2.2}$$

$$(D^{(2)}x)_i := \begin{cases} x_{i+1} - x_i & \text{if } \text{mod}(i,n) \ne 0, \\ x_{i-n+1} - x_i & \text{otherwise.} \end{cases} \tag{2.3}$$

Here, the discrete gradient operators $D^{(1)}$ and $D^{(2)}$ are n^2-by-n^2 matrices, and the i-rows of $D^{(1)}$ and $D^{(2)}$ correspond to the first and second rows of D_i, respectively. The quantity $\|D_i x\|_2$ measures the TV of x at pixel i. The resulting TV is called isotropic. We emphasize that our approach also applies to anisotropic (1-norm) TV deconvolution problems. For simplicity, we will focus on the isotropic case in detail and mention the anisotropic case when necessary.

One fast TV deblurring algorithm for (2.1), called FTVd, was recently proposed in [38]. To make use of the structure of (2.1), the authors first formulate (2.1) as an equivalent constrained problem:

$$\min_{x,y}\left\{\sum_{i=1}^{n^2}\|y_i\|_2+\frac{\mu}{2}\|Kx-f\|_2^2 : y_i=D_ix,\ i=1,\dots,n^2\right\}, \tag{2.4}$$

where $y_i \in \mathbb{R}^2$ is an auxiliary vector. The vector y is defined as

$$y := \begin{pmatrix} y^{(1)} \\ y^{(2)} \end{pmatrix} \in \mathbb{R}^{2n^2} \quad \text{and} \quad y_i := \begin{pmatrix} (y^{(1)})_i \\ (y^{(2)})_i \end{pmatrix} \in \mathbb{R}^2,\ i=1,\dots,n^2 \tag{2.5}$$

(cf. $(D^{(1)}x)_i$ and $(D^{(2)}x)_i$ in definitions (2.2) and (2.3), respectively). Then the authors consider the unconstrained version of (2.4), where the linear constraints are penalized by a quadratic term in the objective function. Finally, an alternate minimization scheme with respect to x and y, together with a continuation scheme on the penalty parameter, is implemented to the unconstrained version. Since every subproblem in each iteration can be solved by either shrinkage or fast Fourier transforms (FFTs), FTVd performs much better than a number of existing methods such as the lagged diffusivity algorithm [37], some Fourier and wavelet shrinkage methods [24], and the MATLAB Image Processing Toolbox functions `deconvwnr` and `deconvreg`. Very recently, an inexact version of ALM was proposed to solve the TV model with nonquadratic fidelity [40], which is also applicable for solving TV-L2 model (2.1).

Another class of algorithms of particular interest is the iterative shrinkage/thresholding (IST)-based algorithms which are proposed and analyzed in different fields [8, 14, 15, 34, 35]. The convergence rate of IST-based algorithms, however, is only $O(k^{-1})$, where k is the number of iterations. There are many efforts to improve its speed, such as the two-step IST (TwIST) algorithm [5] and the fast IST algorithm (FISTA) [3]. In particular, FISTA is inspired by the work of Nesterov [25], and it performs better than the IST and TwIST according to the numerical results reported in [3]. In fact, the authors have shown in [3] that the convergence rate of FISTA is $O(k^{-2})$.

In [4], Beck and Teboulle presented a monotone version of FISTA, called MFISTA, for solving the constrained TV-L2 problem:

$$\min_{x\in\Omega}\left\{\sum_{i=1}^{n^2}\|D_ix\|_2+\frac{\mu}{2}\|Kx-f\|_2^2\right\}. \tag{2.6}$$

As with FISTA, they solve (2.6) by solving a series of denoising problems where the problems are now constrained onto Ω. The constrained denoising problems are transformed into their dual problems and solved by a fast projection gradient method. They showed that MFISTA also has the convergence rate of $O(k^{-2})$. Numerical tests in [4] indicate that by simply imposing the box constraint, the constrained model (2.6) can yield more than 2dB improvement on PSNR for some special images.

Besides the TV-L2 model, another interesting TV deblurring problem is the TV-L1 model, which is good for deblurring images under the corruption of impulse noise [9, 2, 7, 40, 42]:

$$\min_x \left\{ \sum_{i=1}^{n^2} \|D_i x\|_2 + \mu \|Kx - f\|_1 \right\}. \tag{2.7}$$

The developers of FTVd have extended their method to cover this case; see [42]. In addition, an inexact version of ALM was proposed to solve TV-L1 problem (2.7) [40]. To the best of our knowledge, MFISTA has not been extended to (2.7). Also, so far no one has addressed the constrained TV-L1 model,

$$\min_{x\in\Omega} \left\{ \sum_{i=1}^{n^2} \|D_i x\|_2 + \mu \|Kx - f\|_1 \right\}. \tag{2.8}$$

As we have mentioned, in this paper we apply ADMM to solve both the constrained TV-L2 model (2.6) and the constrained TV-L1 model (2.8).

3. Applying ADMM to constrained TV-Lp models. In this section, we apply the ADMM idea to derive algorithms for solving the constrained TV-L2 model (2.6) and the constrained TV-L1 model (2.8). Recall that the basic idea of ADMM goes back to the work of Glowinski and Marrocco [18] and Gabay and Mercier [17], and we refer to some applications in image processing which can be solved by ADMM, e.g., [19, 16, 30, 27, 43, 39, 44, 45, 12].

3.1. Constrained TV-L2 model. To apply the ADMM idea to (2.6), we first introduce two auxiliary variables y and z to change (2.6) to the equivalent form

$$\min_{z\in\Omega,x,y} \left\{ \sum_i \|y_i\|_2 + \frac{\mu}{2}\|Kx - f\|_2^2 : y_i = D_i x,\ i = 1,\dots,n^2; x = z \right\}. \tag{3.1}$$

The auxiliary variable y_i, as defined in (2.5), is to liberate the discrete derivative operator $D_i x$ from the nondifferentiable term $\|\cdot\|_2$, and the variable z plays the role of x within the box constraint so that the box constraint is now imposed on z instead of x. By grouping the variables into two blocks x and (y, z), we see that the objective function of (3.1) is the sum of a function of x and a function of (y, z), and thus ADMM is applicable. In the following, we show that each subproblem of ADMM either has a closed form solution or can be solved by a fast solver.

Let $\mathcal{L}_{\mathcal{A}}(x, y, z; \lambda, \xi)$ be the augmented Lagrangian function of (3.1) which is defined as follows:

$$\begin{aligned}\mathcal{L}_{\mathcal{A}}(x, y, z; \lambda, \xi) \equiv &\sum_i \left(\|y_i\|_2 - \lambda_i^\top (y_i - D_i x) + \frac{\beta_1}{2}\|y_i - D_i x\|_2^2 \right) \\ &+ \frac{\mu}{2}\|Kx - f\|_2^2 - \xi^\top (z - x) + \frac{\beta_2}{2}\|z - x\|_2^2,\end{aligned}$$

where $\beta_1, \beta_2 > 0$ and $\lambda \in \mathbb{R}^{2n^2}$ and $\xi \in \mathbb{R}^{n^2}$ are the Lagrange multipliers. Starting at $x = x^k$, $\lambda = \lambda^k$ and $\xi = \xi^k$, applying ADMM in [18, 17] yields the iterative scheme

$$\begin{pmatrix} y^{k+1} \\ z^{k+1} \end{pmatrix} \leftarrow \arg\min_{z\in\Omega,y} \mathcal{L}_{\mathcal{A}}(x^k, y, z; \lambda^k, \xi^k), \tag{3.2}$$

$$x^{k+1} \leftarrow \arg\min_{x} \mathcal{L}_{\mathcal{A}}(x, y^{k+1}, z^{k+1}; \lambda^k, \xi^k), \tag{3.3}$$

$$\begin{pmatrix} \lambda^{k+1} \\ \xi^{k+1} \end{pmatrix} \leftarrow \begin{pmatrix} \lambda^k - \gamma\beta_1(y^{k+1} - Dx^{k+1}) \\ \xi^k - \gamma\beta_2(z^{k+1} - x^{k+1}) \end{pmatrix}. \tag{3.4}$$

The parameters β_1, β_2 correspond to the linear constraints $y_i = D_i x$ and $x = z$ in (3.1). Theoretically any positive values of β_1 and β_2 ensure the convergence of ADMM [21], and the specific choice of β's we used in the experiments will be specified later.

We now show that the minimization (3.2) with respect to y and z can be separated into two independent subproblems. Firstly, the z-subproblem can be implemented by the simple projection $\mathcal{P}_\Omega$ onto the box:

$$z^{k+1} = \mathcal{P}_\Omega\left[x^k - \frac{\xi^k}{\beta_2}\right]. \tag{3.5}$$

The y-subproblem is equivalent to n^2 two-dimensional problems in the form

$$\min_{y_i\in\mathbb{R}^2}\left\{\|y_i\|_2 + \frac{\beta_1}{2}\left\|y_i - \left(D_i x^k + \frac{1}{\beta_1}(\lambda^k)_i\right)\right\|_2^2\right\}, \quad i = 1, 2, \ldots, n^2. \tag{3.6}$$

According to [38, 41], the solution of (3.6) is given explicitly by the two-dimensional shrinkage:

$$y_i^{k+1} = \max\left\{\left\|D_i x^k + \frac{1}{\beta_1}(\lambda^k)_i\right\|_2 - \frac{1}{\beta_1}, 0\right\}\frac{D_i x^k + \frac{1}{\beta_1}(\lambda^k)_i}{\|D_i x^k + \frac{1}{\beta_1}(\lambda^k)_i\|_2}, \quad i = 1, 2, \ldots, n^2, \tag{3.7}$$

where $0\cdot(0/0) = 0$ is assumed. The computational cost of (3.7) is therefore linear with respect to n^2.

We note that, when the 1-norm is used in the definition of TV, i.e., the TV is anisotropic, y_i^{k+1} will be given by the simpler one-dimensional shrinkage:

$$y_i^{k+1} = \max\left\{\left|D_i x^k + \frac{1}{\beta_1}(\lambda^k)_i\right| - \frac{1}{\beta_1}, 0\right\} \circ \operatorname{sgn}\left(D_i x^k + \frac{1}{\beta_1}(\lambda^k)_i\right), \quad i = 1, 2, \ldots, n^2,$$

where $\circ$ and sgn represent, respectively, the pointwise product and the signum function, and all operations are done componentwise.

Next the minimization (3.3) with respect to x is just a least squares problem, and the corresponding normal equation is

$$\left(D^\top D + \frac{\mu}{\beta_1}K^\top K + \frac{\beta_2}{\beta_1}I\right)x = D^\top\left(y^{k+1} - \frac{1}{\beta_1}\lambda^k\right) + \frac{\mu}{\beta_1}K^\top f + \frac{\beta_2}{\beta_1}\left(z^{k+1} - \frac{\xi^k}{\beta_2}\right), \tag{3.8}$$

where $D \equiv \binom{D^{(1)}}{D^{(2)}} \in \mathbb{R}^{2n^2 \times n^2}$ is the global first-order finite difference operator with $D^{(1)}$ and $D^{(2)}$ being matrices defined by (2.2) and (2.3). Notice that the coefficient matrix in (3.8) is nonsingular whenever β_1, $\beta_2 > 0$. This is an advantage over other splitting methods [38, 41, 42] which, in order to guarantee nonsingularity, require the intersection of the null space of $K^\top K$ and the null space of $D^\top D$ to be the zero vector only. 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., [20, 11]) and thus are diagonalizable by the two-dimensional discrete Fourier transforms (DFTs). As a result, (3.8) can be solved by one forward DFT and one inverse DFT, each at a cost of $O(n^2 \log n)$. If the boundary condition is Neumann and the blur is symmetric, then the coefficient matrix can be diagonalized by discrete cosine transform (DCT) in the same amount of cost; see [26].

Finally, the update (3.4) for λ and ξ can be done straightforwardly in $O(n^2)$ operations.

In conclusion, the main cost per iteration for the scheme (3.2)–(3.4) is dominated by two FFT or DCT operations, and hence is of $O(n^2 \log n)$. Below we give our ADMM-based algorithm for solving the constrained TV-L2 model (2.6).

Algorithm 1. *ADMM for the constrained TV-L2 problem* (2.6).
Input f, K, $\mu > 0$, $\beta_1, \beta_2 > 0$, *and* λ^0. *Initialize* $x = f$ *and* $\lambda = \lambda^0$, $\xi = \xi^0$.
While *"a stopping criterion is not satisfied,"* **Do**
(1) *Compute* y^{k+1} *according to* (3.7).
(2) *Compute* z^{k+1} *according to* (3.5).
(3) *Compute* x^{k+1} *by solving* (3.8).
(4) *Update* λ^{k+1} *and* ξ^{k+1} *via* (3.4).
End Do

Since our method is basically an application of ADMM for the case with two blocks of variables x and (y, z), its convergence is guaranteed by classical results in the ADMM literature, e.g., [6, 17, 18]. We summarize the convergence of Algorithm 1 in the following theorem.

Theorem 3.1. *For* $\beta_1, \beta_2 > 0$ *and* $\gamma \in (0, \frac{1+\sqrt{5}}{2})$, *the sequence* $\{(x^k, y^k, z^k, \lambda^k, \xi^k)\}$ *generated by Algorithm* 1 *from any initial point* (x^0, λ^0, ξ^k) *converges to* $(x^*, y^*, z^*, \lambda^*, \xi^*)$, *where* (x^*, y^*, z^*) *is a solution of* (2.6).

3.2. Constrained TV-L1 model.

In this section, we apply ADMM to solve the constrained TV-L1 model (2.8). Similarly to the constrained TV-L2 case, we introduce three auxiliary variables in (2.8) and transform it into

$$\min_{w \in \Omega, x, y, z} \left\{ \sum_i \|y_i\|_2 + \mu \|z\|_1 : \ y_i = D_i x, \ i = 1, \ldots, n^2, \ z = Kx - f, \ w = x \right\}. \tag{3.9}$$

Note that the constraint is now imposed on w instead of x. The augmented Lagrangian function of (3.9) is

$$\mathcal{L}_\mathcal{A}(x, y, z, w; \lambda, \xi, \zeta) = \sum_i \|y_i\|_2 - \lambda^\top (y - Dx) + \frac{\beta_1}{2} \sum_i \|y_i - D_i x\|_2^2$$

$$+\mu\|z\|_1 - \xi^\top[z-(Kx-f)] + \frac{\beta_2}{2}\|z-(Kx-f)\|_2^2$$
$$-\zeta^\top(w-x) + \frac{\beta_3}{2}\|w-x\|_2^2, \tag{3.10}$$

where $\beta_1,\beta_2,\beta_3 > 0$ and $\lambda\in\mathbb{R}^{2n^2}$, $\xi\in\mathbb{R}^{n^2}$, and $\zeta\in\mathbb{R}^{n^2}$ are the Lagrange multipliers. According to the scheme of ADMM, for a given $(x^k,\lambda^k,\xi^k,\zeta^k)$, the next iterate $(x^{k+1},y^{k+1},z^{k+1},\lambda^{k+1},\xi^{k+1},\zeta^{k+1})$ is generated as follows:

1. Fix $x=x^k$, $\lambda=\lambda^k$, $\xi=\xi^k$, and $\zeta=\zeta^k$, and minimize $\mathcal{L}_\mathcal{A}$ in (3.10) with respect to y, z, and w to obtain y^{k+1}, z^{k+1}, and w^{k+1}. The minimizers are given explicitly by

$$y_i^{k+1} = \max\left\{\left\|D_ix^k + \frac{(\lambda_1)_i^k}{\beta_1}\right\|_2 - \frac{1}{\beta_1}, 0\right\}\frac{D_ix^k + (\lambda_1)_i^k/\beta_1}{\|D_ix^k + (\lambda_1)_i^k/\beta_1\|_2}, \quad i=1,2,\ldots,n^2, \tag{3.11}$$

$$z^{k+1} = \operatorname{sgn}\left(Kx^k - f + \frac{\xi^k}{\beta_2}\right)\circ\max\left\{\left|Kx^k - f + \frac{\xi^k}{\beta_2}\right| - \frac{\mu}{\beta_2, 0}\right\}, \tag{3.12}$$

$$w^{k+1} = P_\Omega\left[x^k + \frac{\zeta^k}{\beta_3}\right], \tag{3.13}$$

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

2. Compute x^{k+1} by solving the normal equation

$$\left(D^\top D + \frac{\beta_2}{\beta_1}K^\top K + \frac{\beta_3}{\beta_1}I\right)x = D^\top\left(y^{k+1} - \frac{\lambda^k}{\beta_1}\right) + \frac{\beta_2}{\beta_1}K^\top\left(z^{k+1} - \frac{\xi^k}{\beta_2}\right) + \frac{\beta_2}{\beta_1}K^\top f + \frac{\beta_3}{\beta_1}\left(w^{k+1} - \frac{\zeta^k}{\beta_3}\right). \tag{3.14}$$

3. Update the multipliers via

$$\begin{cases} \lambda^{k+1} = \lambda^k - \gamma\beta_1(y^{k+1} - Dx^{k+1}),\\ \xi^{k+1} = \xi^k - \gamma\beta_2[z^{k+1} - (Kx^{k+1} - f)],\\ \zeta^{k+1} = \zeta^k - \gamma\beta_3(w^{k+1} - x^{k+1}). \end{cases} \tag{3.15}$$

Next we present an ADMM-based algorithm for solving the constrained TV-L1 model (2.8).

Algorithm 2. *ADMM for the constrained TV-L1 model (2.8).*
Input f, K, $\mu>0$, $\beta_1,\beta_2,\beta_3>0$, and λ^0. Initialize $x=f$ and $\lambda=\lambda^0$, $\xi=\xi^0$, $\zeta=\zeta^0$.
While *"a stopping criterion is not satisfied,"* **Do**
(1) *Compute y^{k+1}, z^{k+1}, and w^{k+1} according to (3.11), (3.12), and (3.13).*
(2) *Compute x^{k+1} by solving (3.14).*
(3) *Update λ^{k+1}, ξ^{k+1}, and ζ^{k+1} via (3.15).*
End Do

Again, Algorithm 2 is an application of ADMM for the case with two blocks of variables (y, z, w) and x. Thus, its convergence is guaranteed by the theory of ADMM, and we summarize it in the following theorem.

Theorem 3.2. *For $\beta_1, \beta_2, \beta_3 > 0$ and $\gamma \in (0, \frac{1+\sqrt{5}}{2})$, the sequence $\{(x^k, y^k, z^k, w^k, \lambda^k, \xi^k, \zeta^k)\}$ generated by Algorithm 1 from any initial point $(x^0, \lambda^0, \xi^0, \zeta^0)$ converges to $(x^*, y^*, z^*, w^*, \lambda^*, \xi^*, \zeta^*)$, where (x^*, y^*, z^*, w^*) is a solution of (2.8).*

4. Numerical experiments. In this section, we apply our algorithms to solve the constrained TV-L2 problem (2.6) and the TV-L1 problem (2.8) and compare them with some state-of-the-art algorithms. The code of our algorithms was written in MATLAB 7.12 (R2011a), and all the numerical experiments were conducted on a ThinkPad notebook with an Intel Core i5-2140M CPU with 2.3 GHz and 4 GB of memory. The quality of our restoration is measured by the peak signal-to-noise ratio (PSNR) in decibel (dB):

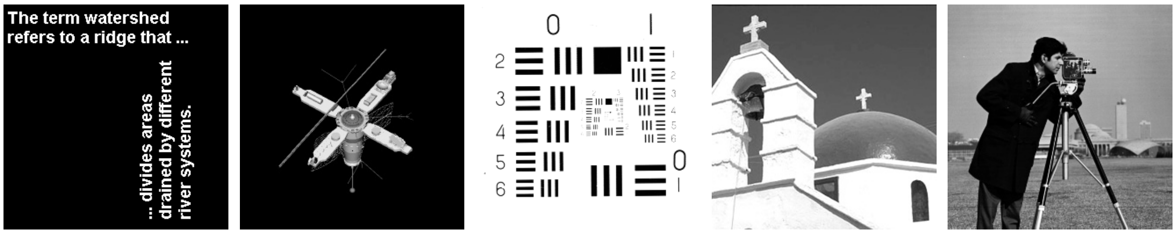
$$\text{PSNR}(x) = 20 \log_{10} \frac{x_{\max}}{\sqrt{\text{Var}(x, \bar{x})}} \quad \text{with} \quad \text{Var}(x, \bar{x}) = \frac{\sum_{j=1}^{n^2} [\bar{x}(j) - x(j)]^2}{n^2}. \tag{4.1}$$

Here $\bar{x}$ is the true image, and $\bar{x}_{\max}$ is the maximum possible pixel value of the image. To make it easier to compare across different models and different methods, we used one uniform stopping criterion for all the algorithms we tested, that is,

$$\frac{|\mathcal{J}^{k+1} - \mathcal{J}^k|}{|\mathcal{J}^k|} < 10^{-5}, \tag{4.2}$$

where $\mathcal{J}^k$ is the objective function value of the respective model in the kth iteration.

The test images are 256-by-256 images as shown in Figure 1: (a) Text.png, (b) Satellite.pgm, (c) Chart.tiff, (d) Church.jpg, and (e) Cameraman.tif. Their pixel values are all scaled to the interval $[0, 1]$ first, so the box constraint in the constrained models is simply $[\mathbf{0}, \mathbf{1}]$ (i.e., $l_i = 0$ and $u_i = 1$ for all i's). We note that the percentages of extreme-value pixels (i.e., pixels with the value 0 or 1) in the five test images are 100%, 89.81%, 84.66%, 22.54%, and 0%, respectively.



(a) Text (b) Satellite (c) Chart (d) Church (e) Cameraman

Figure 1. *Original images.*

One may argue that, for digital images, the pixel values have to be integer too, besides being in a suitable dynamic range. For example, for 8-bit images, the values of x should be integers in [0, 255]. We will see in section 4.3 that, for 8-bit images, the additional

Table 1

Numerical comparison of FTVd, ALM, and Algorithm 1 for images (a)–(e) in Figure 1.

Tested image	Kernel type	Value of μ ($\times 10^4$)			Time (s)			PSNR (dB)						
		FTVd	ALM	Algo. 1	FTVd	ALM	Algo. 1	FTVd	FTVd-S	FTVd-T	ALM	ALM-S	ALM-T	Algo. 1
(a)	I	10	8.6	13	2.25	1.59	1.28	34.65	18.43	35.39	35.22	18.53	35.86	**42.61**
	II	24	21	46	2.45	1.50	15.04	27.49	12.30	28.07	27.67	12.53	28.19	**37.77**
(b)	I	11	10	20	2.17	1.39	13.85	32.92	23.15	33.20	33.02	23.26	33.24	**34.60**
	II	13	11	22	2.17	1.86	14.16	29.87	17.34	30.00	29.95	17.62	30.04	**31.43**
(c)	I	7.1	6.8	14	2.14	1.22	14.49	34.94	20.59	35.27	35.24	20.67	35.53	**37.60**
	II	10	8.9	19	2.95	1.75	13.93	30.80	18.37	31.00	30.99	17.83	31.14	**33.94**
(d)	I	6.1	6.0	6.6	1.87	1.56	1.34	35.20	24.24	35.28	35.27	23.98	35.34	**35.60**
	II	7.8	7.5	8.0	2.11	1.75	1.53	33.04	26.24	33.11	33.08	26.69	33.15	**33.84**
(e)	I	9.3	9.3	9.4	1.70	1.06	1.59	31.54	26.91	31.55	31.52	26.86	31.53	**31.55**
	II	13	13	13	1.97	1.42	1.48	29.42	24.39	29.43	29.38	24.19	29.40	**29.42**

requirement that all pixels be integers in [0,255] affects the restored PSNR values only in the second decimal place; see Tables 4 and 5. Hence in the following experiments, we will not impose integer constraints onto the solution but just the box constraint $[\mathbf{0}, \mathbf{1}]$. In section 4.4, we illustrate that our method is robust against the choice of the regularization parameter μ.

Regarding the penalty parameters β's in Algorithms 1 and 2, theoretically any positive values of β_1 and β_2 ensure the convergence of ADMM [21], and we usually have two ways to determine them in practice. One is to try some values and pick a value with satisfactory performance, and then fix it throughout; the other is to apply some self-adaptive adjustment rules in the literature with an arbitrary initial guess, and this strategy requires no tuning. Since the latter requires expensive computation to realize the self-adaptive adjustments for imaging applications, we used the former strategy. Our experience is that a well-tuned constant value performs almost the same as a self-adaptive strategy.

4.1. Experiments for TV-L2. In this subsection, we focus on the TV deblurring problem with Gaussian noise. Our purposes are (i) to show the accuracy of the constrained TV-L2 model (2.6) over the unconstrained model, and (ii) to demonstrate the efficiency of the proposed Algorithm 1. The efficiency of Algorithm 1 is shown mainly by comparison with the FTVd in [38], an inexact version of ALM (see Algorithm 4.2 in [40]), and the MFISTA in [4]. The inexact version of ALM (i.e., Algorithm 4.2 in [40]) coincides with the split Bregman algorithm in [19]. In this way, we have also compared with the split Bregman algorithm in [19]. Note that Algorithm 4.2 in [40] requires an inner iteration to solve the primal variables alternatively. In [40], the authors mentioned that "the split Bregman method will waste the accuracy of the inner iteration and does not speed up dramatically when the inner iteration number $L > 1$." Therefore, in the numerical tests of [40], the authors simply set the inner iteration number $L = 1$. Thus, we also set $L = 1$ in our comparison. In the following, we denote Algorithm 4.2 in [40] as ALM.

4.1.1. Comparison with FTVd and ALM. We first compare our Algorithm 1 with FTVd and ALM (see Table 1). Recall that FTVd and ALM solve the unconstrained model (2.1), while Algorithm 1 tackles the constrained model (2.6). As we have mentioned, one can solve the unconstrained TV-L2 model (2.1) first and then project the restored image onto the box constraint by either the *truncation* or the *stretching* procedure. Therefore we report the result of FTVd and ALM with these two minimize-and-project procedures (denoted, respectively, by FTVd-T and FTVd-S, and ALM-T and ALM-S), in addition to the original FTVd and ALM.

The ten blurred and noisy images in our tests are degraded as follows. Since the periodic boundary condition is used to generate the convolution operator in [38, 40], we use the same boundary condition to blur the test images. Two types of blurring kernels are tested: type I (`fspecial ('average',9)`), and type II (`fspecial ('gaussian',[9,9],3)`). For each case, the blurred images are further corrupted by Gaussian noise with zero mean and standard deviation of size 10^{-3}.

We use the code provided by the authors of [38] to implement the FTVd (but with the stopping criterion (4.2)). Thus, the values of all involved parameters of FTVd remain unchanged. The code for ALM was coded by us. For the penalty parameter r_p required in this algorithm, we tuned and set it to 10 as it gives satisfactory speed for all 10 test cases. For our Algorithm 1, we tuned and set $\beta_1 = 10$, $\beta_2 = 20$, and $\gamma = 1.618$ as they give satisfactory speed for all 10 test cases, too. All iterations start with the degraded images. Finally, as is well known, the accuracy of the solution depends on the value of the regularization parameter μ. We thus tuned it manually (up to two significant digits) and chose the one that gave the highest PSNR value. The values of μ for each algorithm are listed in Table 1. The computing time in seconds (Time (s)) to satisfy the stopping criterion (4.2) and the PSNR values of the restored images are also reported in Table 1. The best method for each test case is highlighted in boldface. In Figure 2 we display the restored Satellite images from different algorithms.

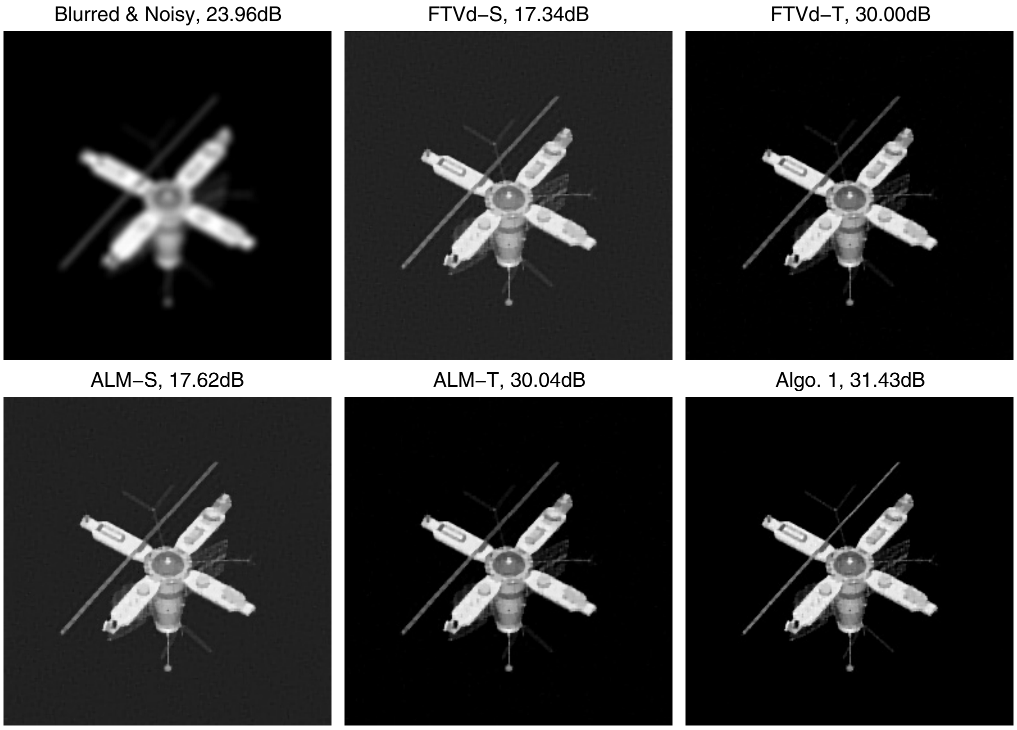


Figure 2. *Top row: blurred and noisy image (left) and restored images of FTVd-S (middle), FTVd-T (right). Bottom row: ALM-S (left), ALM-T (middle), and Algorithm* 1 *(right) (type* II *blur).*

Some observations can be made based on the results in Table 1 and Figure 2.

- Our constrained TV-L2 model (2.6) is particularly accurate for deblurring images with a high percentage of extreme-value pixels; see, e.g., the test images (a), (b), and (c) in Figure 1. In particular, for image (a) Text blurred by type II noise, the restored PSNR by Algorithm 1 is 10.28dB higher than that by FTVd, 25.47dB higher than that by "FTVd-S," and 9.70dB higher than that by "FTVd-T"; it is 10.10dB higher than that by "ALM," 25.24dB higher than that by "ALM-S," and 9.58dB higher than that by "ALM-T." For the more realistic satellite image (b), the gain is still at least 1.36dB for type I blur and 1.39dB for type II blur.
- For images whose percentages of extreme-value pixels are not high, such as the test images (d) and (e) in Figure 1, the model (2.6) is as accurate as (2.1) with the gain in PSNR values being small but still positive. The gain is at least 0.26dB for image (d) and 0.02dB for image (e)—recall that image (e) has no extreme-value pixels. We remark that in these cases, Algorithm 1 performs competitively with ALM in terms of CPU time, and both methods are faster than FTVd.
- The *truncation* procedure is much more accurate than the *stretching* procedure, especially for images with a high percentage of extreme-value pixels. This can be explained from the last two plots in the first row and the first two plots in the second row of Figure 2, where we see that stretching changes the contrast of the images and hence the value of every pixel in the images.
- If one solves the models by ALM or FTVd and then projects by truncation, one can increase the PSNR. This means that the solutions of ALM and FTVd do not automatically satisfy the box constraints.

4.1.2. Comparison with MFISTA. In this subsection, we compare Algorithm 1 with MFISTA in [4] for TV-L2 deblurring problems. Recall that MFISTA also solves the constrained TV-L2 problem (2.6). Following [4], we use the Neumann boundary condition to generate the blur, where the kernel size is 9-by-9. The blurred images are further corrupted by Gaussian noise with zero mean and standard deviation of size 2-by-10^{-2}.

We use the original MFISTA code provided by the authors of [4] but with the stopping criterion (4.2). MFISTA applies a fast projection gradient (FPG) method to solve the constrained model (2.6). At each FPG step, it is necessary to solve a constrained denoising problem. The authors apply the same FPG method to solve the dual form of (2.6), and a fixed number of 10 FPG steps is recommended in [4]. We followed their suggestion and used 10 FPG steps here. To implement Algorithm 1, we take $\beta_1 = \beta_2 = 0.01$ and $\gamma = 1.618$. As in section 4.1.1, we tune the value of μ manually to two significant digits, and the best values are listed in Table 2. In Table 2, we also report the computing time in seconds, the restored PSNR, and the objective function value (obj-end) when the stopping criterion (4.2) is satisfied. Table 2 shows that Algorithm 1 can restore images with the same quality as those by MFISTA (which is not surprising as the two algorithms are solving the same constrained TV-L2 model) but with a much faster speed. To illustrate this more clearly, in Figure 3, we depict the objective function value and the PSNR value with respect to the computing time for image (a). Clearly our method converges much faster than MFISTA. Similar curves and conclusions can be drawn for the other four test images.

Table 2
Numerical comparison of MFISTA and Algorithm 1 for model (2.6).

Tested image	Value of μ ($\times 10^3$) in (2.6)	Time (s)		PSNR (dB)		obj-end	
		MFISTA	Algo. 1	MFISTA	Algo. 1	MFISTA	Algo. 1
(a)	6.3	23.56	6.19	20.52	**20.59**	26.16	26.12
(b)	1.0	31.34	6.10	26.91	**26.92**	27.12	27.10
(c)	2.3	24.57	7.32	23.18	**23.20**	28.25	28.22
(d)	0.9	19.62	2.85	26.98	**27.01**	28.23	28.23
(e)	1.0	21.82	5.72	**24.39**	**24.39**	27.72	27.72

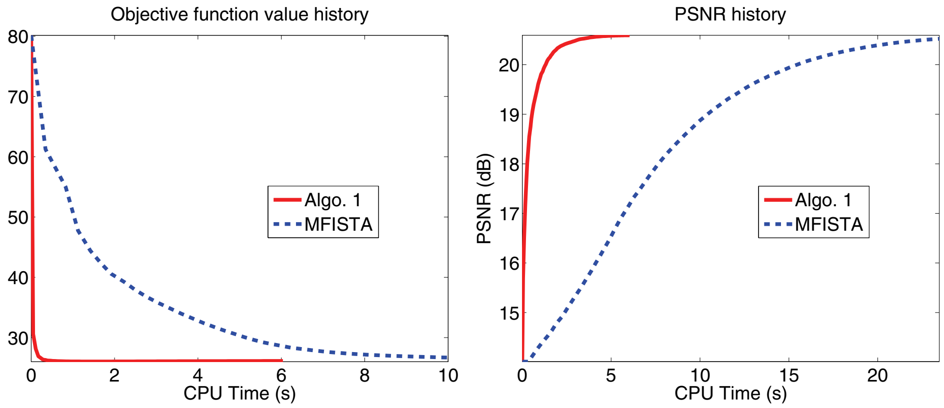


Figure 3. *Objective function value history (left) and PSNR history (right) with respect to the CPU time.*

4.2. Experiments for TV-L1. In this subsection, we focus on the TV deblurring problem with impulse noise. Our purposes are (i) to show the accuracy of the constrained TV-L1 model (2.8) over the unconstrained alternative (2.7), and (ii) to demonstrate the efficiency of Algorithm 2, mainly via a comparison with FTVd [42] and ALM [40].

Fifteen degraded test images are generated in a way similar to that in section 4.1. That is, we first generated the blurred image with the periodic boundary condition and then corrupted the blurred images by salt-and-pepper impulse noise with the noise level 40%, 50%, and 60%. The blurring operator is the Gaussian blur used in [42] which has a kernel size of 7-by-7 and standard deviation 5. Again, we use the original code of FTVd but with the stopping criterion (4.2). To get a best performance, we set $r_p = 5$, and $r_z = 20$ in ALM (see [40]). In Algorithm 2, we set $\beta_1 = 5$, $\beta_2 = 20$, $\beta_3 = 10$, $\gamma = 1.618$. We tune μ manually for each algorithm, and their best values are listed in Table 3. FTVd and ALM with the *truncation* or *stretching* projection are also compared.

Conclusions similar to those in section 4.1.1 can be made based on the results in Table 3. For example, when compared to the unconstrained model (2.7) with or without a projection procedure, the constrained TV-L1 model (2.8) is always more accurate, with a possible improvement of more than 2.06dB in PSNR (see image (a) with a 40% level of noise). Again, this superiority is more obvious for images with higher percentages of extreme-value pixels. In addition, Algorithm 2 outperforms FTVd in speed for all cases. Also the *truncation* procedure is again more accurate than the *stretching* procedure.

Table 3
Numerical comparison of FTVd, ALM, and Algorithm 2.

Image	Noise level	Value of μ			Time (s)			PSNR (dB)						
		FTVd	ALM	Algo. 2	FTVd	ALM	Algo. 2	FTVd	FTVd-S	FTVd-T	ALM	ALM-S	ALM-T	Algo. 2
(a)	40%	21	35	55	4.38	3.63	4.01	19.53	9.75	19.64	21.33	7.98	21.68	**23.74**
	50%	12	25	50	5.07	2.39	4.21	16.64	11.81	16.65	18.37	7.53	18.62	**20.17**
	60%	3.7	16	28	6.69	1.73	2.42	15.08	14.62	15.08	16.23	8.66	16.31	**17.04**
(b)	40%	25	23	24	4.71	2.20	2.20	28.67	18.52	28.71	28.04	23.20	28.07	**28.45**
	50%	25	23	24	4.68	1.33	2.01	27.57	20.60	27.60	27.46	24.63	27.48	**27.92**
	60%	7.2	10	10	5.91	1.19	1.54	26.75	25.13	26.76	26.90	24.94	26.93	**27.06**
(c)	40%	17	32	36	5.66	4.18	5.13	23.94	15.88	23.98	26.46	13.89	26.66	**27.59**
	50%	11	21	33	5.55	3.43	4.31	20.42	13.74	20.46	22.86	213.81	23.05	**24.19**
	60%	6	14	21	5.77	2.07	3.37	18.01	13.86	18.03	19.51	12.89	19.65	**20.79**
(d)	40%	17	20	17	4.63	2.98	3.92	30.81	24.31	30.84	30.62	23.40	30.68	**31.16**
	50%	12	14	17	5.02	2.07	3.70	28.55	20.98	28.58	28.98	21.75	29.01	**29.44**
	60%	7.8	12	12	6.41	1.95	2.65	26.16	19.71	26.17	26.91	18.83	26.95	**27.22**
(e)	40%	11	25	25	4.38	3.03	4.12	25.78	25.58	25.78	26.55	20.20	26.57	**26.60**
	50%	5.9	17	20	4.60	1.98	3.14	24.37	22.32	24.37	25.43	22.72	25.44	**25.50**
	60%	2.9	12	11	5.99	1.78	2.61	22.97	20.54	22.97	24.23	22.10	24.23	**24.23**

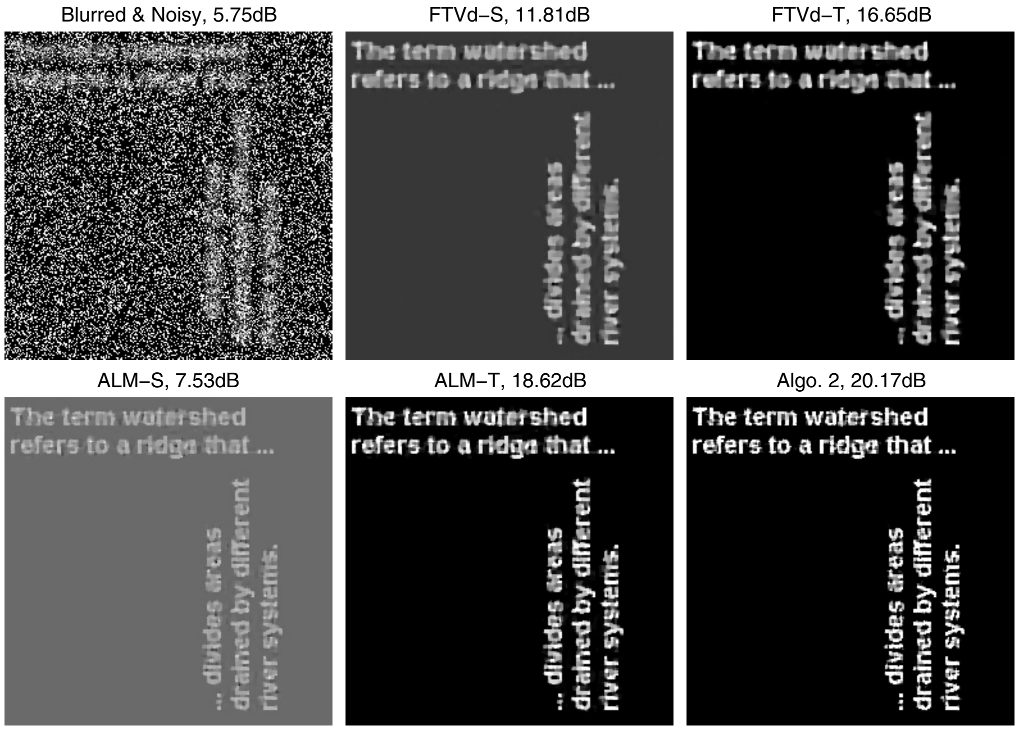


Figure 4. *Top row: blurred and noisy image (left) and restored images of FTVd-S (middle), FTVd-T (right). Bottom row: ALM-S (left), ALM-T (middle), and Algorithm 2 (right).*

Finally, in Figure 4, we display the degraded image and the restored text images for 50% level of noise by the three methods. We can easily see visual improvement in the image by using our method.

4.3. Integer constraints. One may remark that for digital images, the pixel values have to be integer, too, besides being in the dynamic range $[l, u]$. We repeated the experiments in

Table 4

PSNR comparison when integer constraints are added to Table 1.

Tested image	Kernel type	FTVd-S		FTVd-T		ALM-S		ALM-T		Algo. 2	
		$\mathbb{R}$	$\mathbb{Z}$	$\mathbb{R}$	$\mathbb{Z}$	$\mathbb{R}$	$\mathbb{Z}$	$\mathbb{R}$	$\mathbb{Z}$	$\mathbb{R}$	$\mathbb{Z}$
(a)	I	18.43	18.43	35.39	35.38	18.53	18.53	35.86	35.85	**42.61**	42.58
	II	12.30	12.30	28.07	28.06	12.53	12.53	28.19	28.19	37.77	**37.78**
(b)	I	23.15	23.14	33.20	33.20	23.26	23.25	33.24	33.24	**34.60**	**34.60**
	II	17.34	17.34	30.00	30.00	17.62	17.62	30.04	30.04	**31.43**	**31.43**
(c)	I	20.59	20.59	35.27	35.26	20.67	20.68	35.53	35.52	**37.60**	37.59
	II	18.37	18.37	31.00	30.99	17.83	17.83	31.14	31.14	**33.94**	**33.94**
(d)	I	24.24	24.24	35.28	35.27	23.98	23.98	35.34	35.32	**35.60**	35.59
	II	26.24	26.24	33.11	33.11	26.69	26.69	33.15	33.14	**33.84**	33.83
(e)	I	26.91	26.91	31.55	31.54	26.86	26.86	31.53	31.52	**31.55**	31.54
	II	24.39	24.39	29.43	29.43	24.19	24.19	29.40	29.39	**29.42**	**29.42**

Table 5

PSNR comparison when integer constraints are added to Table 3.

Tested image	Noise level	FTVd-S		FTVd-T		ALM-S		ALM-T		Algo. 1	
		$\mathbb{R}$	$\mathbb{Z}$	$\mathbb{R}$	$\mathbb{Z}$	$\mathbb{R}$	$\mathbb{Z}$	$\mathbb{R}$	$\mathbb{Z}$	$\mathbb{R}$	$\mathbb{Z}$
(a)	40%	9.75	9.78	19.64	19.64	7.98	7.98	21.68	21.67	**23.74**	**23.74**
	50%	11.81	11.79	16.65	16.65	7.53	7.52	18.62	18.62	**20.17**	**20.17**
	60%	14.62	14.62	15.08	15.08	8.66	8.68	16.31	16.31	**17.04**	**17.04**
(b)	40%	18.52	18.53	28.71	28.71	23.20	23.17	28.07	28.06	**28.45**	**28.45**
	50%	20.60	20.63	27.60	27.60	24.63	24.58	27.48	27.48	**27.92**	**27.92**
	60%	25.13	25.11	26.76	26.76	24.94	26.93	26.93	26.93	**27.06**	**27.06**
(c)	40%	15.88	15.85	23.98	23.98	13.89	13.86	26.66	26.66	**27.59**	**27.59**
	50%	13.74	13.76	20.46	20.46	13.81	13.84	23.05	23.05	**24.19**	**24.19**
	60%	13.86	13.88	18.03	18.03	12.89	12.87	19.65	19.65	**20.79**	**20.79**
(d)	40%	24.31	24.30	30.84	30.84	23.40	23.41	30.68	30.68	**31.16**	**31.16**
	50%	20.98	20.97	28.58	28.57	21.75	21.75	29.01	29.01	**29.44**	**29.44**
	60%	19.71	19.71	26.17	26.17	18.83	18.83	26.95	26.95	**27.22**	**27.22**
(e)	40%	25.58	25.58	25.78	25.77	20.20	20.20	26.57	26.57	**26.60**	26.59
	50%	22.32	22.31	24.37	24.37	22.72	22.72	25.44	25.44	**25.50**	**25.50**
	60%	20.54	20.54	22.97	22.97	22.10	22.10	**24.23**	**24.23**	**24.23**	**24.23**

Tables 1 and 3, but this time we scaled and rounded the pixel values of the restored images to integers in [0,255]. Tables 4 and 5 give the resulting PSNR. In these tables, $\mathbb{R}$ means we do not impose the integer constraints, and all restored images are in [0,1], while $\mathbb{Z}$ means we have imposed the integer constraints. From the tables we see that the integer constraints affect the PSNR values only in the second decimal place. The results justify our consideration of imposing only the box constraint $[\mathbf{0}, \mathbf{1}]$.

4.4. Robustness to μ. In this subsection, we point out an important advantage of our algorithms. So far the numerical results have been presented based on the fact that the regularization parameter μ is tuned manually, with the purpose of maximizing the PSNR value of the restored image. Here we compare the constrained and unconstrained models simultaneously for a wide range of possible μ, as shown in Figure 5. For the sake of succinctness, we give the comparison only for the TV-L1 problem; the results for the TV-L2 problem are analogous. Since it has been shown that the FTVd-S and ALM-S are significantly worse than

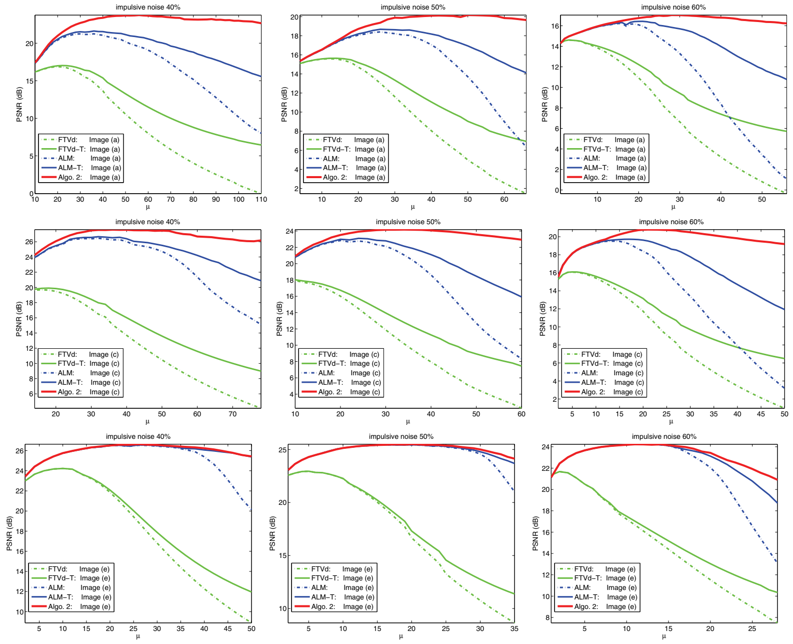


Figure 5. *Restored PSNR with respect to μ for FTVd, FTVd-T, ALM, ALM-T, and Algorithm* 1. *Row* 1*, image* (a)*; row* 2*, image* (c)*; row* 3*, image* (e)*. Noise level: column* 1*,* 40%*; column* 2*,* 50%*; column* 3*,* 60%*.*

the FTVd-T and ALM-T, we focus only on the comparison of FTVd, FTVd-T, ALM, ALM-T, and Algorithm 2. Recall that Algorithm 2 solves the constrained model (2.8), while FTVd and ALM both solve the unconstrained model (2.7).

In Figure 5, we plot the restored PSNR values for FTVd, FTVd-T, ALM, ALM-T, and Algorithm 2 against different values of μ. Each column in Figure 5 corresponds to the three cases with three different impulsive noise levels (40%, 50%, and 60%). The stopping criterion is still (4.2). For the sake of simplicity, only images (a), (c), and (e) with noise levels 40%, 50%, and 60% are shown. Figure 5 further verifies that box constraints in TV deblurring models can improve the restoration quality significantly. In fact, for all μ, our Algorithm 2 always gives higher PSNR values than FTVd, FTVd-T, ALM, and ALM-T. This advantage is more obvious for an image with many extreme-value pixels (e.g., image (a)). Moreover, the PSNR curves of Algorithm 2 are flatter than the curves from the other four methods, showing that our Algorithm 2 is a good method for a wider range of μ. However, one may get very bad restoration from FTVd if a larger than optimal μ is chosen, as its PSNR curves dive rapidly after they peak. Thus, the constrained TV models are more robust than their unconstrained counterparts with respect to the changes in the regularization parameter μ.

5. Concluding remarks. In this paper, we show the necessity of considering box constraints in total variation (TV) models for image deblurring problems. We discuss the cases of both Gaussian and impulse noise and propose accordingly the box-constrained TV-L2 and TV-L1 models. We demonstrate that the constrained TV models can easily be solved by the alternating direction method of multipliers (ADMM). Two fast ADMM-based algorithms are thus developed for solving the constrained TV-L2 and TV-L1 models. The accuracy of our proposed constrained TV models (compared to unconstrained TV models) and the efficiency of the ADMM-based algorithms (compared to some state-of-the-art methods) are verified by numerical examples. Clearly if the true images do not have many pixels that are on the boundary of the constraints (e.g., the Cameraman image has none), then one may prefer not to use our method as it will waste time in enforcing the box constraints. However, the numerical results show that the overhead is not big even in those cases.

REFERENCES

[1] L. Ambrosio and V. Tortorelli, *Approximation of functionals depending on jumps by elliptic functionals via* Γ*-convergence*, Comm. Pure Appl. Math., 43 (1990), pp. 999–1036.

[2] L. Bar, A. Brook, N. Sochen, and N. Kiryati, *Deblurring of color images corrupted by salt-and-pepper noise*, IEEE Trans. Image Process., 16 (2007), pp. 1101–1111.

[3] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[4] A. Beck and M. Teboulle, *Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems*, IEEE Trans. Image Process., 18 (2009), pp. 2419–2434.

[5] 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 (2007), pp. 2992–3004.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., 3 (2011), pp. 1–122.

[7] J.-F. Cai, R. H. Chan, and M. Nikolova, *Fast two-phase image deblurring under impulse noise*, J. Math. Imaging Vision, 36 (2010), pp. 46–53.

[8] J. F. Cai, R. H. Chan, and Z. W. Shen, *A framelet-based image inpainting algorithm*, Appl. Comput. Harmon. Anal., 24 (2008), pp. 131–149.

[9] R. H. Chan, C. W. Ho, and M. Nikolova, *Salt-and-pepper noise removal by median-type noise detectors and edge-preserving regularization*, IEEE Trans. Image Process., 14 (2005), pp. 1479–1485.

[10] R. H. Chan, B. Morini, and M. Porcelli, *Affine scaling methods for image deblurring problems*, AIP Conf. Proc., 1281 (2010), pp. 1043–1046.

[11] R. H. Chan and M. K. Ng, *Conjugate gradient method for Toeplitz systems*, SIAM Rev., 38 (1996), pp. 427–482.

[12] R. H. Chan, J. Yang, and X. Yuan, *Alternating direction method for image inpainting in wavelet domains*, SIAM J. Imaging Sci., 4 (2011), pp. 807–826.

[13] S. Durand, J. Fadili, and M. M. Nikolova, *Multiplicative noise cleaning via a variational method involving curvelet coefficients*, in Scale Space and Variational Methods in Computer Vision, Lecture Notes in Comput. Sci. 5567, X.-C. Tai, K. Morken, M. Lysaker, and K.-A. Lie, eds., Springer, Berlin, 2009, pp. 282–294.

[14] M. Elad, *Why simple shrinkage is still relevant for redundant representations?*, IEEE Trans. Inform. Theory, 52 (2006), pp. 5559–5569.

[15] M. Elad, B. Matalon, and M. Zibulevsky, *Image denoising with shrinkage and redundant representations*, in Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, 2006.

[16] E. Esser, *Applications of Lagrangian-based Alternating Direction Methods and Connections to Split Bregman*, UCLA CAM Report 09-31, UCLA, Los Angeles, CA, 2009.

[17] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appl., 2 (1976), pp. 17–40.

[18] R. Glowinski and A. Marrocco, *Sur l'approximation par éléments finis d'ordre un, et la résolution par pénalisation-dualité d'une classe de problemes de Dirichlet non linéaires*, RAIRO Anal. Numér., 9 (1975), pp. 41–76.

[19] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.

[20] R. Gonzalez and R. Woods, *Digital Image Processing*, Addison-Wesley, Reading, MA, 1992.

[21] B. S. He and H. Yang, *Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities*, Oper. Res. Lett., 23 (1998), pp. 151–161.
[22] B. Morini, M. Porcelli, and R. H. Chan, *A reduced Newton method for constrained linear least-squares problems*, J. Comput. Appl. Math., 233 (2010), pp. 2200–2212.
[23] D. Mumford and J. Shah, *Optimal approximations by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42 (1989), pp. 577–685.
[24] R. Neelamani, H. Choi, and R. G. Baraniuk, *Wavelet-based deconvolution for ill-conditioned systems*, in Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 6, 1999, pp. 3241–3244.
[25] Y. E. Nesterov, *A method for solving the convex programming problem with convergence rate* $O(1/k^2)$, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543–547 (in Russian).
[26] 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.
[27] M. K. Ng, P. Weiss, and X. Yuan, *Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods*, SIAM J. Sci. Comput., 32 (2010), pp. 2710–2736.
[28] L. Rudin and S. Osher, *Total variation based image restoration with free local constraints*, in Proceedings of the IEEE International Conference on Image Processing, Vol. 1, 1994, pp. 31–35.
[29] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[30] S. Setzer, *Split Bregman Algorithm, Douglas-Rachford splitting and frame shrinkage*, in Scale Space and Variational Methods in Computer Vision, Lecture Notes in Comput. Sci. 5567, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 464–476.
[31] J. Shah, *A common framework for curve evolution, segmentation and anisotropic diffusion*, in Proceedings of the 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1996, pp. 136–142.
[32] J. Shi and S. Osher, *A nonlinear inverse scale space method for a convex multiplicative noise model*, SIAM J. Imaging Sci., 1 (2008), pp. 294–321.
[33] Y. Shi, L.-L. Wang, and X.-C. Tai, *Geometry of total variation regularized* L_p*-model*, J. Comput. Appl. Math., 236 (2012), pp. 2223–2234.
[34] J. L. Starck, E. Donoho, and D. Candès, *Astronomical image representation by the curvelet transform*, Astron. Astrophys., 398 (2003), pp. 785–800.
[35] J. L. Starck, M. Nguyen, and F. Murtagh, *Wavelets and curvelets for image deconvolution: A combined approach*, Signal Process., 83 (2003), pp. 2279–2283.
[36] A. Tikhonov and V. Arsenin, *Solution of Ill-Posed Problems*, V. H. Winston & Sons, Washington, DC, 1977.
[37] C. R. Vogel and M. E. Oman, *Iterative methods for total variation denoising*, SIAM J. Sci. Comput., 17 (1996), pp. 227–238.
[38] Y. Wang, J. Yang, W. Yin, and Y. Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imaging Sci., 1 (2008), pp. 248–272.
[39] C. Wu and X.-C. Tai, *Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models*, SIAM J. Imaging Sci., 3 (2010), pp. 300–339.
[40] C. Wu, J. Zhang, and X.-C. Tai, *Augmented Lagrangian method for total variation restoration with non-quadratic fidelity*, Inverse Probl. Imaging, 5 (2011), pp. 237–261.
[41] J. Yang, W. Yin, Y. Zhang, and Y. Wang, *A fast algorithm for edge-preserving variational multichannel image restoration*, SIAM J. Imaging Sci., 2 (2009), pp. 569–592.
[42] 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.
[43] J. Yang, Y. Zhang, and W. Yin, *A fast alternating direction method for* $TVL1$-$L2$ *signal reconstruction from partial Fourier data*, IEEE J. Sel. Topics Signal Process., 4 (2010), pp. 288–297.
[44] X. Zhang, M. Burger, X. Bresson, and S. Osher, *Bregmanized nonlocal regularization for deconvolution and sparse reconstruction*, SIAM J. Imaging Sci., 3 (2010), pp. 253–276.
[45] X. Q. Zhang, M. Burger, and S. Osher, *A unified primal-dual algorithm framework based on Bregman iteration*, J. Sci. Comput., 46 (2010), pp. 20–46.