SIAM J. SCI. COMPUT.
Vol. 31, No. 4, pp. 2842–2865

AN EFFICIENT TVL1 ALGORITHM FOR DEBLURRING MULTICHANNEL IMAGES CORRUPTED BY IMPULSIVE NOISE*

JUNFENG YANG†, YIN ZHANG‡, AND WOTAO YIN‡

Abstract. We extend the alternating minimization algorithm recently proposed in [Y. Wang, J. Yang, W. Yin, and Y. Zhang, *SIAM J. Imag. Sci.*, 1 (2008), pp. 248–272]; [J. Yang, W. Yin, Y. Zhang, and Y. Wang, *SIAM J. Imag. Sci.*, 2 (2009), pp. 569–592] to the case of recovering blurry multichannel (color) images corrupted by impulsive rather than Gaussian noise. The algorithm minimizes the sum of a multichannel extension of total variation and a data fidelity term measured in the ℓ_1-norm, and is applicable to both salt-and-pepper and random-valued impulsive noise. We derive the algorithm by applying the well-known quadratic penalty function technique and prove attractive convergence properties, including finite convergence for some variables and q-linear convergence rate. Under periodic boundary conditions, the main computational requirements of the algorithm are fast Fourier transforms and a low-complexity Gaussian elimination procedure. Numerical results on images with different blurs and impulsive noise are presented to demonstrate the efficiency of the algorithm. In addition, it is numerically compared to the least absolute deviation method [H. Y. Fu, M. K. Ng, M. Nikolova, and J. L. Barlow, *SIAM J. Sci. Comput.*, 27 (2006), pp. 1881–1902] and the two-phase method [J. F. Cai, R. Chan, and M. Nikolova, *AIMS J. Inverse Problems and Imaging*, 2 (2008), pp. 187–204] for recovering grayscale images. We also present results of recovering multichannel images.

Key words. impulsive noise, cross-channel, image deblurring, isotropic total variation, fast Fourier transform

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

DOI. 10.1137/080732894

1. Introduction.

We consider the problem of recovering multichannel images degraded by cross-channel blurring and impulsive noise (e.g., salt-and-pepper noise). Without loss of generality, we assume that the underlying images have square domains and let an $n \times n$ original image with m channels be denoted by $\bar{u} = (\bar{u}^{(1)}; \dots ; \bar{u}^{(m)}) \in \mathbb{R}^{mn^2}$, where $\bar{u}^{(j)} \in \mathbb{R}^{n^2}$ represents the jth channel, $j = 1, \dots, m$. The observation $f \in \mathbb{R}^{mn^2}$ of $\bar{u}$ is

$$f = \mathbb{N}_{\text{imp}}(K\bar{u}), \tag{1.1}$$

where $\mathbb{N}_{\text{imp}}(\cdot)$ models the corruption of impulsive noise and

$$K = \begin{bmatrix} K_{11} & K_{12} & \dots & K_{1m} \\ K_{21} & K_{22} & \dots & K_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ K_{m1} & K_{m2} & \dots & K_{mm} \end{bmatrix} \in \mathbb{R}^{mn^2 \times mn^2} \tag{1.2}$$

*Received by the editors August 13, 2008; accepted for publication (in revised form) March 30, 2009; published electronically July 3, 2009.
http://www.siam.org/journals/sisc/31-4/73289.html

†Department of Mathematics, Nanjing University, 22 Hankou Road, Nanjing, Jiangsu Province, 210093, People's Republic of China (jfyang2992@gmail.com). The work of this author was supported by the Chinese Scholarship Council during his visit to Rice University.

‡Department of Computational and Applied Mathematics, Rice University, 6100 Main Street, MS-134, Houston, Texas 77005 (yin.zhang, wotao.yin@rice.edu). The work of the second author was supported by NSF Grant DMS-0811188. The work of the third author was supported by NSF Career Grant DMS-0748839.

is a cross-channel blurring operator. We note that each K_{ij} is a convolution matrix with certain block structures determined by boundary conditions (specified later). Given K, our objective is to recover $\bar{u}$ from the blurry and noisy observation f. We perform deblurring and denoising jointly by solving a multichannel total variation (TV) regularization problem with the 1-norm data fidelity:

$$\min_u \sum_{i=1}^{n^2} \|(I_m \otimes D_i)u\|_2 + \mu\|Ku - f\|_1, \tag{1.3}$$

where I_m is the identity matrix of order m, "$\otimes$" represents the Kronecker product, and $(I_m \otimes D_i)u \in \mathbb{R}^{2m}$ consists of certain first-order horizontal and vertical finite differences of u at pixel i. In particular, for red-green-blue (RGB) color images where $m = 3$, there hold

$$u = \begin{pmatrix} u^{(r)} \\ u^{(g)} \\ u^{(b)} \end{pmatrix} \in \mathbb{R}^{3n^2} \quad \text{and} \quad (I_3 \otimes D_i)u = \begin{pmatrix} D_i u^{(r)} \\ D_i u^{(g)} \\ D_i u^{(b)} \end{pmatrix} \in \mathbb{R}^6.$$

Formulation (1.3) is often referred to as a TV regularization with the 1-norm fidelity (TVL1) model. More generally, our algorithm applies to the following local weighted TVL1-like problem:

$$\min_u \sum_{i=1}^{n^2} \alpha_i \|G_i u\|_2 + \mu\|Ku - f\|_1, \tag{1.4}$$

where, at each pixel i, $G_i u \in \mathbb{R}^q$ (e.g., $q = 2m$ for multichannel TV as in (1.3)) represents a certain local finite difference of u over all channels, $\alpha_i > 0$ is a weighting parameter, and $\mu > 0$ balances the regularization term and the fidelity term. We note that second- and higher-order finite differences are permitted in the regularization term of (1.4). Although μ can be removed from (1.4) by rescaling α_i, $i = 1, \ldots, n^2$, we keep it for convenience. As such, problem (1.4) reduces to (1.3) by letting $G_i = I_m \otimes D_i$ and $\alpha_i \equiv 1$. The origin of (1.3) and some related results are reviewed briefly in subsection 1.2.

The main contribution of this paper is an efficient algorithm for solving (1.4). Our algorithm can be derived from either the classic quadratic penalty function technique in optimization, dated back to Courant's work [19] in 1943, or the half-quadratic technique initially proposed by Geman and Reynolds in [28] and Geman and Yang in [29]. In this paper, our derivation follows the former technique for its simplicity. A derivation based on a smoothed approximation of TV and the additive form of half-quadratic technique [29] can be found in [51] for a deblurring model under Gaussian noise.

It is well known that, for a fixed $t \in \mathbb{R}$, the scalar minimizer of $|x| + |x - t|^2$ is given by soft thresholding. In [50, 51], this result was extended to the multidimensional case. Specifically, for any positive integer q and $\mathbf{t} \in \mathbb{R}^q$, the minimizer of $\|\mathbf{s}\|_2 + \|\mathbf{s} - \mathbf{t}\|_2^2$ is given by

$$\mathbf{s} = \max\left\{\|\mathbf{t}\|_2 - \frac{1}{2}, 0\right\} \frac{\mathbf{t}}{\|\mathbf{t}\|_2}, \tag{1.5}$$

where the convention $0 \cdot (0/0) = 0$ is followed. We call the operation in (1.5) multidimensional shrinkage, which plays an important role in this paper. In order to utilize

shrinkage operations, we propose to solve (1.4) by the classical quadratic penalty method. Let $z \in \mathbb{R}^{mn^2}$ be an auxiliary variable that approximates $Ku - f$ in (1.4). Similarly, at each pixel i, we introduce $\mathbf{w}_i \in \mathbb{R}^q$ to approximate $G_i u \in \mathbb{R}^q$. For convenience, we let $\mathbf{w} = [\mathbf{w}_1, \ldots, \mathbf{w}_{n^2}]$. Then, by adding quadratic terms to penalize the difference between every pair of original and auxiliary variables, we obtain the following approximation problem to (1.4):

$$\min_{\mathbf{w},z,u} \sum_i \left(\alpha_i \|\mathbf{w}_i\| + \frac{\beta_1}{2} \|\mathbf{w}_i - G_i u\|^2 \right) + \mu \|z\|_1 + \frac{\beta_2}{2} \|z - (Ku - f)\|^2, \tag{1.6}$$

where $\beta_1, \beta_2 \gg 0$ are penalty parameters. We introduce (1.6) because it is numerically easier to minimize by an iterative and alternating approach due to the fact that with any two of the three variables $\mathbf{w}$, z, and u fixed, the minimizer of (1.6) with respect to the third one has a closed-form formula that is easy to compute. In addition, this approach is numerically stable for large values of β_1 and β_2. Since $\mathbf{w}$ and z are decoupled for given u, our algorithm will minimize the objective function in (1.6) with respect to $(\mathbf{w}, z)$ and u, alternately. We show that, for any fixed $\beta_1, \beta_2 > 0$, this alternating minimization scheme generates a sequence of points converging to a solution of (1.6). Besides, we establish finite convergence for some auxiliary variables and fast q-linear convergence for the rest. Furthermore, the overall convergence is significantly accelerated by a continuation approach on the penalty parameters. From the well-known theory of penalty method, the solutions of (1.6) converge to that of (1.4) as the penalty parameters go to infinity. The convergence behavior of such penalty methods as applied to TVL2 models (where a 2-norm square is used for the fidelity term) has been empirically studied in [50, 51]. Our experiments indicate that the convergence behavior of solutions of (1.6) to that of (1.4) should be similar to the one found in [50, 51].

Although derived for $\|G_i u\|_2$ in (1.4), the algorithm can be easily modified for $\|G_i u\|_1$, or more generally, $h(G_i u)$ for a convex function $h(\cdot)$. In what follows, we first give a brief review of impulsive noise removal methods, as well as variational approaches for image denoising and/or deblurring, and then summarize the contributions and organization of this paper.

1.1. Impulsive noise and removal methods. Two common types of impulsive noise are salt-and-pepper noise, corrupting a portion of all pixels with minimal or maximal intensities and leaving the remainder unaffected, and random-valued noise, the same as salt-and-pepper noise except that corrupted pixels have random intensity values between minimal and maximal ones. Such noise is often generated by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or erroneous transmission; see, e.g., [6]. In images contaminated by such noise, a certain number of pixels of the underlying image are uncorrupted, and the corrupted pixels usually have intensities distinguishable from those of their neighbors. Based on these characters, various nonlinear digital filter methods have been proposed; see [2]. Among them, the median-type filters are most popular due to their good denoising power and computational efficiency, e.g., the adaptive median filter [30], the multistate median filter [18], and the median filter based on homogeneity information [23, 41]. These filters first detect possibly noisy data entries and then replace them by using the median filters or their variants. Most of these filters were designed for denoising only and are not suitable for deblurring.

1.2. Variational approach. Another important class of methods for removing impulsive noise is the variational approach. It is well known that recovering $\bar{u}$ from f by inverting (1.1) is ill posed because the solution is highly sensitive to the noise. To stabilize the recovery of $\bar{u}$, regularization is introduced, giving rise to the variational formulation

$$\min_u \Phi_{\text{reg}}(u) + \mu\Phi_{\text{fid}}(u, f), \tag{1.7}$$

where the *regularization term* $\Phi_{\text{reg}}(u)$ models some a priori information about $\bar{u}$, the *fidelity term* $\Phi_{\text{fid}}(u, f)$ measures some type of deviation of u from the observation f, and $\mu > 0$ balances these two terms in the formulation.

Traditional regularization techniques include the Tikhonov-like regularization [48], the TV regularization [42], both of which have been well studied for grayscale images, and others. A discrete Tikhonov-like regularization takes the form $\Phi_{\text{reg}}(u) = \sum_i \sum_j \|(D^{(j)}u)_i\|_2^2$, 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 resultant minimization problems are relatively easy to solve, Tikhonov-like regularization tends to make images overly smooth and often fails to adequately preserve important image attributes such as sharp edges. In comparison, TV regularization overcomes these drawbacks. The discrete form of TV for a grayscale image $u \in \mathbb{R}^{n^2}$ is given by $\text{TV}(u) = \sum_i \|D_i u\|$. If $\|\cdot\|$ is the 2-norm, $\text{TV}(u)$ is isotropic because it is irrelevant to the rotation of data in this case. In other cases, e.g., $\|\cdot\| = \|\cdot\|_1$, it is anisotropic. We point out that, although the isotropic TV is often preferred over any anisotropic one, both types of discretizations lead to the so-called staircasing effects; see, e.g., [22, 37]. Compared with Tikhonov-like regularization, TV regularization has the advantage of preserving sharp edges and object boundaries.

For multichannel images, it is important to couple channels in regularization. For this purpose, TV has been extended to "color TV" in [5] and multichannel TV in [8, 13, 14, 47], which is represented by the first term in (1.3) where at each pixel the 2-norm measures variations from all channels. Another approach of coupling channels in regularization is the Beltrami flow proposed in [45], where a multichannel image is treated as a surface in higher dimensional space and then the area of the surface was minimized. In [31], the authors applied the Beltrami flow to color image denoising and blind deconvolution. In our algorithm, we use multichannel TV.

In the literature, the common data fidelity for the Gaussian noise is $\Phi_{\text{fid}}(u, f) = \|Ku - f\|_2^2$, which also represents the maximum likelihood estimation of $\bar{u}$. Such data fidelity is used in denoising, deblurring, and various inverse problems; see, e.g., [48, 42, 49]. However, practical systems suffer from *outliers* such as salt-and-pepper noise, where only a portion of data entries are corrupted by noise of some non-Gaussian distribution. In such cases, minimizing $\Phi_{\text{fid}}(u, f) = \|Ku - f\|_2^2$ will fail to preserve the uncorrupted data entries [38], while minimizing nonsmooth data fidelity will likely succeed. Theoretical results comparing smooth and nonsmooth data fidelity terms for image denoising can be found in [38, 39]. It is pointed out in [38] that nonsmooth fidelity terms can give correct pixel intensities under some favorable conditions.

The use of nonsmooth fidelity $\Phi_{\text{fid}}(u, f) = \|Ku - f\|_1$ was introduced in [1], where a pure denoising problem was considered. Recently, geometric properties of the TVL1 model, which uses TV with the above ℓ_1-fidelity term, were analyzed in [10]. Also, it was shown in [21] that TVL1 minimization yields a contrast invariant and self-dual filter. Besides, the authors of [54, 55] proved that the TVL1 model has some

interesting properties that lead to multiscale decomposition, contrast preservation, and morphological invariance. They also established the equivalence between the TVL1 model and certain geometric optimization problems, which was used to show that this model decomposes an image (or any signal) into components of different scales, measured in terms of G-value [44], independent of their locations or intensities. These results have led to the applications of the TVL1 model in medical imaging [53, 15] and computer vision [17, 16], in which this model has shown its advantages over some competing models because it requires no feature or structural information beside spacial scales.

Recently, a two-phase approach was proposed in [11, 12] for denoising and in [9] for deblurring where impulsive plus Gaussian noise is permitted. The deblurring in [9] is performed in two stages. In the first stage, a set of likely outliers is identified using median filters and removed from the data set. In the second stage, the image is restored using a specialized regularization method that applies to the remaining data entries.

At present, highly efficient numerical methods are still in need for solving variational models that use a nonsmooth fidelity term for image deblurring. For TV-based deblurring models using smooth fidelity terms, the majority of existing algorithms are based on solving the Euler–Lagrangian equations by gradient descent or fixed-point iterations; see, e.g., [49, 5]. For TVL1 problem (1.3), similar methods are used in [3]. Compared to our proposed algorithm that takes advantage of fast transforms, the existing methods are slow, especially when the size of the blurring kernel is relatively large [50].

1.3. Contributions. The main contribution of this paper is an efficient algorithm for solving the general TVL1-like problem (1.4), which encompasses a variety of regularization functions such as weighted TV and those based on higher-order derivatives. In addition, this algorithm is analyzed and shown to have attractive convergence properties, which include global convergence with a strong q-linear rate and finite convergence for some auxiliary variables. Under periodic boundary conditions, its computation can take advantage of simple multidimensional shrinkage and fast Fourier transform (FFT).

1.4. Organization. The paper is organized as follows. In section 2, the alternating minimization algorithm is derived, and the optimality conditions of (1.4) and (1.6) are studied. In section 3, the main convergence results of the proposed algorithm are established. Practical implementation of the proposed algorithm and comparisons to the methods proposed in [26] and [9] are given in section 4. In this section, multichannel image recovery results are also presented. Finally, concluding remarks are given in section 5.

2. Basic algorithm and optimality. Before deriving the algorithm, we introduce our notation. We let the superscript $^\top$ be the transpose (or conjugate transpose) operation for real (or complex) quantities. Let $D^{(1)}, D^{(2)} \in \mathbb{R}^{n^2 \times n^2}$ be the first-order forward finite difference matrices in horizontal and vertical directions, respectively. Without loss of generality, we assume that $(I_m \otimes D_i)u$ consists of the first-order forward finite differences of u at pixel i in (1.3). Thus, $D_i \in \mathbb{R}^{2\times n^2}$ is a two-row matrix formed by stacking the ith row of $D^{(1)}$ on that of $D^{(2)}$. For vectors v_1 and v_2, we let $(v_1; v_2)$ be the vector formed by stacking the two. Let $\rho(T)$ be the spectral radius of matrix T. Hereafter, the norm $\|\cdot\|$ refers to the 2-norm. Additional notation will be introduced as the paper progresses.

2.1. An alternating minimization algorithm. We begin with assuming $\beta_1 = \beta_2 \equiv \beta$ in (1.6), which does not cause any loss of generality because, otherwise, a scaling of the form $\mu \leftarrow \mu\sqrt{\beta_1/\beta_2}$ and $(K, f, z) \leftarrow \sqrt{\beta_2/\beta_1}(K, f, z)$ would equalize the penalty parameters in (1.6). We derive our algorithm in this section and analyze its convergence in section 3 under this assumption even though β_1 and β_2 are set to be different in our implementation. It is easy to see that for a fixed u, the minimization with respect to $\mathbf{w}$ and z can be done in parallel because they are separable in (1.6). In addition, for all subscripts i, the first two terms in (1.6) are separable with respect to $\mathbf{w}_i$ and the the last two terms are separable with respect to each component of z. Based on these observations, it is easy to apply alternating minimization to (1.6). First, for a fixed u, the minimizer function of $\mathbf{w}_i$ is given by a multidimensional shrinkage similar to the one in (1.5):

$$\mathbf{w}_i = \max\left\{\|G_i u\| - \frac{\alpha_i}{\beta}, 0\right\}\frac{G_i u}{\|G_i u\|},\ i = 1, \ldots, n^2, \tag{2.1}$$

where we followed the convention $0 \cdot (0/0) = 0$, and the minimization with respect to z is given by the well-known one-dimensional shrinkage:

$$z = \max\left\{|Ku - f| - \frac{\mu}{\beta}, 0\right\} \circ \operatorname{sgn}(Ku - f), \tag{2.2}$$

where "$\circ$" represents the pointwise product, and all other operations are implemented componentwise. Clearly, the computational cost for (2.1) is linear in terms of qn^2, and that for (2.2) is linear in terms of n^2. Second, for fixed $\mathbf{w}$ and z, the minimization with respect to u is a least squares problem:

$$\min_u \sum_i \|\mathbf{w}_i - G_i u\|^2 + \|Ku - (f + z)\|^2. \tag{2.3}$$

Let $G^{(j)} \in \mathbb{R}^{n^2 \times mn^2}$ be the matrix formed by stacking the jth rows of $G_1, G_2, \ldots, G_{n^2}$, $j = 1, \ldots, q$,

$$G \triangleq \begin{pmatrix} G^{(1)} \\ \vdots \\ G^{(q)} \end{pmatrix} \in \mathbb{R}^{qn^2 \times mn^2} \text{ and } \mathbf{W} \triangleq \begin{pmatrix} \mathbf{w}_1^\top \\ \vdots \\ \mathbf{w}_{n^2}^\top \end{pmatrix} \triangleq (w_1, \ldots, w_q) \in \mathbb{R}^{n^2 \times q}, \tag{2.4}$$

where w_j is the jth column of $\mathbf{W}$ and formed by stacking the jth components of $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{n^2}$. Let $w = \mathbf{W}(:) = (w_1; \ldots; w_q) \in \mathbb{R}^{qn^2}$, which is the vectorization of $\mathbf{W}$ column by column. For example, when $m = 3$ and $G_i = I_3 \otimes D_i$, it holds that

$$G = \begin{pmatrix} G^{(1)} \\ G^{(2)} \\ G^{(3)} \\ G^{(4)} \\ G^{(5)} \\ G^{(6)} \end{pmatrix} = \begin{pmatrix} D^{(1)} & \mathbf{0} & \mathbf{0} \\ D^{(2)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & D^{(1)} & \mathbf{0} \\ \mathbf{0} & D^{(2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & D^{(1)} \\ \mathbf{0} & \mathbf{0} & D^{(2)} \end{pmatrix}.$$

In the above notation, the normal equations of (2.3) can be written as

$$\left(G^\top G + K^\top K\right) u = G^\top w + K^\top (z + f). \tag{2.5}$$

Since each $G^{(j)}$ is a finite difference matrix applied to some channel, and K defined in (1.2) is a cross-channel convolution, under the periodic boundary conditions for u, each $n^2 \times n^2$ block in G and K has a block circulant structure, while each block within the structure is a circulant matrix; see, e.g., [36]. Therefore, both G and K can be blockwise diagonalized by pre- and postmultiplying by

$$\mathbf{F}_p \triangleq I_p \otimes \mathcal{F}, \quad p = m, q,$$

and their inverses $\mathbf{F}_p^\top = \mathbf{F}_p^{-1} = I_p \otimes \mathcal{F}^{-1}$, respectively, where $\mathcal{F}$ represents the two-dimensional discrete Fourier transform matrix of order $n^2 \times n^2$. More precisely, the two matrices

$$\hat{G} = \mathbf{F}_q G \mathbf{F}_m^\top \quad \text{and} \quad \hat{K} = \mathbf{F}_m K \mathbf{F}_m^\top$$

are both blockwise diagonal matrices with the block size $n^2 \times n^2$. Then (2.5) is equivalent to

$$\left(\hat{G}^\top \hat{G} + \hat{K}^\top \hat{K}\right)(\mathbf{F}_m u) = \mathbf{F}_m \left(G^\top w + K^\top (z + f)\right), \tag{2.6}$$

where $\mathbf{F}_m v$ is the discrete Fourier transform of v. Since both $\hat{G}$ and $\hat{K}$ are blockwise diagonal, the coefficient matrix in the left-hand side of (2.6) is also blockwise diagonal with $m \times m$ blocks of the size $n^2 \times n^2$.

The solution of (2.3) can be obtained by solving its normal equations (2.5) in three steps. First, we apply discrete FFTs to both sides of (2.5). Then, we solve the resulting blockwise diagonal systems (2.6) by Gaussian elimination for $\mathbf{F}_m u$. We note that, in general, a linear system involving such a blockwise diagonal matrix can be solved by block Gaussian elimination without any pivoting or fillings, though for large multichannel images, the cost of such a block Gaussian elimination can still be relatively high in comparison to most other operations required by the algorithm. Finally, we apply $\mathbf{F}_m^{-1}$ to $\mathbf{F}_m u$ to obtain a new iterate u. Clearly, the blockwise diagonal matrices $\hat{G}$ and $\hat{K}$ need to be computed only once before iteration. The total number of two-dimensional discrete Fourier transforms (including inverse Fourier transforms) of size $n^2 \times n^2$ is $2m$ per iteration. Moreover, block Gaussian elimination applied to (2.6) without pivoting, where the coefficient matrix has $m \times m$ blocks of square diagonal matrices of size n^2, requires about $(2/3)m^3 n^2$ arithmetic operations.

Alternatively, under the Neumann boundary conditions and assuming that all blurring kernels are central symmetric, the FFTs shall be replaced by the discrete cosine transforms (DCTs); see [36]. In our experiments, we assumed periodic boundary conditions and used FFTs.

Since minimizing the objective function in (1.6) with respect to each variable is computationally inexpensive, we propose solving (1.6) by the following alternating minimization scheme.

ALGORITHM 1. *Input f, K, $\mu > 0$, $\beta \gg 0$, and $\alpha_i > 0, i = 1, \ldots, n^2$. Initialize $u = f$.*

While *"not converged,"* **Do**

1) *Given u, compute* $\mathbf{w}$ *and z by* (2.1) *and* (2.2)*, respectively.*
2) *Given* $\mathbf{w}$ *and z, compute u by solving* (2.5)*.*

End Do

The stopping criterion of Algorithm 1 is specified in the next subsection based on the optimality conditions of (1.6). More details of Algorithm 1 are discussed in section 4.

2.2. Optimality conditions. Now, we derive optimality conditions of (1.4) and (1.6) and specify a stopping criterion for Algorithm 1. We need the following propositions.

PROPOSITION 2.1. *For any $A \in \mathbb{R}^{p\times n}$, the subdifferential of $f(x) \triangleq \|Ax\|$ is*

$$\partial f(x) = \begin{cases} \left\{A^\top Ax/\|Ax\|\right\}, & \text{if } Ax \neq 0; \\ \left\{A^\top h : \|h\| \leq 1, h \in \mathbb{R}^p\right\}, & \text{otherwise.} \end{cases} \tag{2.7}$$

The proof of Lemma 2.1 is elementary and thus is omitted. For $t \in \mathbb{R}$, the signum and the signum set-valued functions are defined as

$$\operatorname{sgn}(t) \triangleq \begin{cases} +1 & t > 0, \\ 0 & t = 0, \\ -1 & t < 0, \end{cases} \quad \text{and} \quad \operatorname{SGN}(t) \triangleq \begin{cases} \{+1\} & t > 0, \\ [-1,1] & t = 0, \\ \{-1\} & t < 0, \end{cases}$$

respectively. For vector $v \in \mathbb{R}^N$, we let $\operatorname{sgn}(v) = (\operatorname{sgn}(v_1); \ldots ; \operatorname{sgn}(v_N)) \in \mathbb{R}^N$, where v_i is the ith component of v. Similarly, $\operatorname{SGN}(v) = \left\{\xi \in \mathbb{R}^N : \xi_i \in \operatorname{SGN}(v_i), \forall i\right\}$.

PROPOSITION 2.2. *For any $B \in \mathbb{R}^{m\times n}$, the subdifferential of $g(x) \triangleq \|Bx\|_1$ is*

$$\partial g(x) = \left\{B^\top \lambda : \lambda \in \operatorname{SGN}(Bx)\right\}. \tag{2.8}$$

Proof. By the definition of subdifferential for a convex function, we have

$$\partial g(x) = \left\{\xi \in \mathbb{R}^n : \|By\|_1 - \|Bx\|_1 \geq \xi^\top (y - x), \forall y\right\}.$$

We will show that $\partial g(x) = S \triangleq \{B^\top \lambda : \lambda \in \operatorname{SGN}(Bx)\}$. First, for any $B^\top \lambda \in S$ and $y \in \mathbb{R}^n$, it holds that

$$\begin{aligned} &\|By\|_1 - \|Bx\|_1 - \left(B^\top \lambda\right)^\top (y - x) \\ &\qquad = \sum_i \left\{|(By)_i| - |(Bx)_i| - \lambda_i[(By)_i - (Bx)_i]\right\}. \end{aligned} \tag{2.9}$$

Since $\lambda \in \operatorname{SGN}(Bx)$, it is easy to argue that the above is always nonnegative. Thus, $S \subset \partial g(x)$. Next, we show $\partial g(x) \subset S$ by contradiction. Suppose there exists $\xi \in \partial g(x)$, but $\xi \notin S$. Since S is closed and convex, by the well-known separation, theorem of convex sets, there must exist $\eta \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$ such that the hyperplane $\eta^\top x = \alpha$ separates ξ and $\mathcal{S}$ so that $\eta^\top \xi > \alpha > \eta^\top (B^\top \lambda), \forall \lambda \in \operatorname{SGN}(Bx)$. Let $I^+ = \{i : (Bx)_i > 0\}$, $I^- = \{i : (Bx)_i < 0\}$, and $I^0 = \{i : (Bx)_i = 0\}$. It follows that

$$\begin{aligned} &\eta^\top \xi > \alpha \geq \sup\left\{\eta^\top \left(B^\top \lambda\right) : \lambda \in \operatorname{SGN}(Bx)\right\} \\ &\qquad = \sum_{i\in I^+} (B\eta)_i - \sum_{i\in I^-} (B\eta)_i + \sum_{i\in I^0} |(B\eta)_i|. \end{aligned} \tag{2.10}$$

Let $y = x + \epsilon\eta$ for some $\epsilon > 0$ small enough. Since $\xi \in \partial g(x)$, we get

$$\begin{aligned} \epsilon\xi^\top \eta \leq \|By\|_1 - \|Bx\|_1 &= \sum_{i\in I^+\cup I^-} (|(Bx)_i + \epsilon(B\eta)_i| - |(Bx)_i|) + \sum_{i\in I^0} \epsilon|(B\eta)_i| \\ &= \sum_{i\in I^+} \epsilon(B\eta)_i - \sum_{i\in I^-} \epsilon(B\eta)_i + \sum_{i\in I^0} \epsilon|(B\eta)_i|. \end{aligned}$$

The above inequality contradicts (2.10). Therefore, $\partial g(x) = S$ and the result is proved. □

Since the objective function is convex, a triplet $(\mathbf{w}, z, u)$ is a solution of (1.6) if and only if the subdifferential of the objective at $(\mathbf{w}, z, u)$ contains the origin. First, for fixed z and u, this condition implies

$$\mathbf{0} \in \alpha_i \partial \|\mathbf{w}_i\| + \beta\left(\mathbf{w}_i - G_i u\right).$$

In light of Proposition 2.1 with A being the identity of order q, the above condition is equivalent to

$$\begin{cases} \alpha_i \mathbf{w}_i / \|\mathbf{w}_i\| + \beta(\mathbf{w}_i - G_i u) = 0, & i \in I_1 \triangleq \{i : \mathbf{w}_i \neq \mathbf{0}\}, \\ \beta \|G_i u\| \leq \alpha_i, & i \in I_2 \triangleq \{i : \mathbf{w}_i = \mathbf{0}\}. \end{cases} \tag{2.11}$$

Similarly, for fixed $\mathbf{w}$ and u, the optimality condition for z is

$$\mathbf{0} \in \mu \partial \|z\|_1 + \beta\left[z - (Ku - f)\right],$$

which, in light of Proposition 2.2 with B being the identity of order mn^2, is equivalent to

$$\begin{cases} \mu \operatorname{sgn}(z_i) + \beta[z_i - (Ku - f)_i] = 0, & i \in I_3 \triangleq \{i : z_i \neq 0\}, \\ \beta |(Ku - f)_i| \leq \mu, & i \in I_4 \triangleq \{i : z_i = 0\}. \end{cases} \tag{2.12}$$

Finally, for fixed $\mathbf{w}$ and z, the objective function in (1.6) is differentiable with respect to u. The corresponding optimality condition for u is given by

$$G^\top (Gu - w) + K^\top (Ku - f - z) = 0. \tag{2.13}$$

We note that (2.13) is just another expression of (2.5). Let K_{I_3}, K_{I_4} be the submatrices of K containing rows corresponding to those indices in I_3 and I_4, respectively. Similarly, v_I represents the subvector of v corresponding to indices in I. From (2.11), (2.12), and the relation between G and G_i described in (2.4), eliminating $\mathbf{w}$ and z from (2.13) gives

$$\sum_{i \in I_1} \alpha_i G_i^\top \frac{G_i u}{\|G_i u\|} + \sum_{i \in I_2} G_i^\top h_i + \mu \begin{pmatrix} K_{I_3} \\ K_{I_4} \end{pmatrix}^\top \begin{pmatrix} \operatorname{sgn}(K_{I_3} u - f_{I_3}) \\ v_{I_4} \end{pmatrix} = 0, \tag{2.14}$$

where $h_i \triangleq \beta G_i u$ satisfies $\|h_i\| \leq \alpha_i$ and $v_{I_4} = \beta(K_{I_4} u - f_{I_4})/\mu$ satisfies $\|v_{I_4}\|_\infty \leq 1$. Let u^* be any solution of (1.4). Define

$$\begin{aligned} &I_1^* \triangleq \{i : G_i u^* \neq 0\}, \;\; I_2^* \triangleq \{i : G_i u^* = 0\}, \\ &I_3^* \triangleq \{i : (Ku^* - f)_i \neq 0\} \;\text{ and }\; I_4^* \triangleq \{i : (Ku^* - f)_i = 0\}. \end{aligned}$$

In light of Propositions 2.1 and 2.2, there exist $\|h_i^*\| \leq \alpha_i$, $i \in I_2^*$, and $v_{I_4^*} = \{v_i : |v_i| \leq 1, i \in I_4^*\}$ such that

$$\sum_{i \in I_1^*} \alpha_i G_i^\top \frac{G_i u^*}{\|G_i u^*\|} + \sum_{i \in I_2^*} G_i^\top h_i^* + \mu \begin{pmatrix} K_{I_3^*} \\ K_{I_4^*} \end{pmatrix}^\top \begin{pmatrix} \operatorname{sgn}\left(K_{I_3^*} u^* - f_{I_3^*}\right) \\ v_{I_4^*} \end{pmatrix} = 0. \tag{2.15}$$

Equation (2.14) differs from (2.15) only in the index sets involved. As β increases, I_1 and I_3 should approach I_1^* and I_3^*, respectively. Comparing (2.14) and (2.15) provides another perspective on how (1.6) is approximating (1.4).

Our stopping criterion for Algorithm 1 is based on (2.11), (2.12), and (2.13). Let

$$\begin{cases} r_1(i) \triangleq (\alpha_i \mathbf{w}_i/\|\mathbf{w}_i\|)/\beta + \mathbf{w}_i - G_i u, & i \in I_1, \\ r_2(i) \triangleq \|G_i u\| - \alpha_i/\beta, & i \in I_2, \\ r_3(i) \triangleq \mu\,\mathrm{sgn}(z_i)/\beta + z_i - (Ku - f)_i, & i \in I_3, \\ r_4(i) \triangleq |(Ku - f)_i| - \mu/\beta, & i \in I_4, \\ r_5 \triangleq G^\top(Gu - w) + K^\top(Ku - f - z), \end{cases}$$

where $\{I_j : j = 1, 2, 3, 4\}$ are defined as in (2.11) and (2.12). According to the optimality conditions given in (2.11) and (2.12), all components of r_2 and r_4 are nonpositive at a solution. In our implementation, we allow slightly positive elements in r_2 and r_4 at an approximate solution. Specifically, given a tolerance $\epsilon > 0$, Algorithm 1 is terminated once all of the following criteria are met:

$$\begin{cases} \max\{\|r_1(i)\| : i \in I_1\} \le \epsilon, \\ \max\{r_2(i) : i \in I_2\} \le \epsilon, \\ \max\{|r_3(i)| : i \in I_3\} \le \epsilon, \\ \max\{r_4(i) : i \in I_4\} \le \epsilon, \\ \|r_5\| \le \epsilon \cdot \|G^\top w + K^\top(f + z)\|, \end{cases} \tag{2.16}$$

where, for given w and z, the last line in (2.16) measures the relative error of (2.13).

3. Convergence analysis. As pointed out before, the quadratic penalty method applied to a problem like (1.4) converges to its solution as the penalty parameter goes to infinity (see Theorem 17.1 in [40], for example). In this section, we establish convergence and a q-linear convergence result of the proposed algorithm for fixed β.

For simplicity, we assume that $\alpha_i \equiv 1$ and all analysis below can be easily extended to the case $\alpha_i \not\equiv 1$. First, we introduce some notation. For $t \in \mathbb{R}$, let the one-dimensional shrinkage be defined by

$$s_\mu(t) = \max\left\{|t| - \frac{\mu}{\beta}, 0\right\} \cdot \mathrm{sgn}(t).$$

For $v \in \mathbb{R}^N$, let $s_\mu(v) \triangleq (s_\mu(v_1); \ldots; s_\mu(v_N)) \in \mathbb{R}^N$; i.e., s_μ applies to each component v_i of v. For $\mathbf{t} \in \mathbb{R}^q$, let the q-dimensional shrinkage be defined by

$$s(\mathbf{t}) = \max\left\{\|\mathbf{t}\| - \frac{1}{\beta}, 0\right\} \frac{\mathbf{t}}{\|\mathbf{t}\|},$$

where $0 \cdot (0/0) = 0$ is followed. For vectors $v_1, \ldots, v_q \in \mathbb{R}^N$, $S(v_1; \ldots; v_q) : \mathbb{R}^{qN} \to \mathbb{R}^{qN}$ is defined as

$$\begin{aligned} S(v_1; \ldots; v_q) &\triangleq (s(\mathbf{t}_1); \ldots; s(\mathbf{t}_N)), \quad \text{where} \\ \mathbf{t}_i &= ((v_1)_i; \ldots; (v_q)_i) \in \mathbb{R}^q,\ i = 1, \ldots, N. \end{aligned} \tag{3.1}$$

Let $\mathcal{P}(\cdot) \triangleq \mathcal{P}_{\mathcal{B}}(\cdot) : \mathbb{R}^q \to \mathbb{R}^q$ be the projection onto the closed ball $\mathcal{B} \triangleq \{\mathbf{t} \in \mathbb{R}^q : \|\mathbf{t}\| \le 1/\beta\}$. The following lemma shows that $s(\cdot)$ is nonexpansive. As a corollary, both S and s_μ are nonexpansive.

LEMMA 3.1. *For any $\mathbf{t}_1, \mathbf{t}_2 \in \mathbb{R}^q$, it holds that*

$$\|s(\mathbf{t}_1) - s(\mathbf{t}_2)\|^2 \le \|\mathbf{t}_1 - \mathbf{t}_2\|^2 - \|\mathcal{P}(\mathbf{t}_1) - \mathcal{P}(\mathbf{t}_2)\|^2.$$

Furthermore, if $\|s(\mathbf{t}_1) - s(\mathbf{t}_2)\| = \|\mathbf{t}_1 - \mathbf{t}_2\|$, then $s(\mathbf{t}_1) - s(\mathbf{t}_2) = \mathbf{t}_1 - \mathbf{t}_2$.

Proof. The proof of Lemma 3.1 is similar to the case when $q = 2$ in [50]. ∎

The analysis below is accomplished under the following mild assumption, which has been commonly used in previous works of similar analysis.

Assumption 1. $\mathcal{N}(G) \cap \mathcal{N}(K) = \{0\}$, where $\mathcal{N}(\cdot)$ represents the null space of its argument.

The following matrices are used in our analysis:

$$H = (G; K) \quad \text{and} \quad M = G^\top G + K^\top K = H^\top H.$$

Under Assumption 1, M^{-1} is well defined. Furthermore, let $v = (w; z) \in \mathbb{R}^{(q+m)n^2}$ and define

$$h^{(j)}(v) = G^{(j)} M^{-1} \left(H^\top v + K^\top f \right), \quad j = 1, \ldots, q,$$

and

$$h^{(q+1)}(v) = K M^{-1} \left(H^\top v + K^\top f \right) - f.$$

Let $h(v) = (h^{(1)}(v); \ldots; h^{(q)}(v))$, $\hat{h}(v) = (h(v); h^{(q+1)}(v))$, and $\hat{S} \circ \hat{h} = (S \circ h; s_\mu \circ h^{(q+1)})$.

Using the above notation, the iteration formulas (2.1), (2.2), and (2.5) can be expressed as

$$v^{k+1} = \left(w^{k+1}; z^{k+1}\right) = \left(S\left(G^{(1)} u^k; \ldots; G^{(q)} u^k \right); s_\mu \left(K u^k - f \right) \right) = \hat{S} \circ \hat{h} \left(v^k \right) \tag{3.2}$$

and

$$u^{k+1} = M^{-1} \left(H^\top v^{k+1} + K^\top f \right). \tag{3.3}$$

Since the objective function in (1.6) is convex, bounded below, and coercive (i.e., its value goes to infinity as $\|(w, z, u)\| \to \infty$), problem (1.6) has at least one minimizer $(v^*; u^*) = (w^*; z^*; u^*)$, which should satisfy the fixed-point equations

$$v^* = (w^*; z^*) = \left(S\left(G^{(1)} u^*; \ldots; G^{(q)} u^* \right); \quad s_\mu \left(K u^* - f \right) \right) = \hat{S} \circ \hat{h} \left(v^* \right), \tag{3.4}$$

and the equation

$$u^* = M^{-1} \left(H^\top v^* + K^\top f \right). \tag{3.5}$$

To establish convergence of Algorithm 1, we need the following lemmas.

LEMMA 3.2. *For any $v_1 \neq v_2$ in $\mathbb{R}^{(q+m)n^2}$, it holds that*

$$\left\| \hat{h}(v_1) - \hat{h}(v_2) \right\| \leq \|v_1 - v_2\|$$

with the equality holding if and only if $\hat{h}(v_1) - \hat{h}(v_2) = v_1 - v_2$.

LEMMA 3.3. *Let v^* be any fixed point of $\hat{S} \circ \hat{h}$. For any v, we have $\|\hat{S} \circ \hat{h}(v) - \hat{S} \circ \hat{h}(v^*)\| < \|v - v^*\|$ unless v is a fixed point of $\hat{S} \circ \hat{h}$.*

Lemma 3.2 shows that $\hat{h}$ is nonexpansive and Lemma 3.3 gives a useful property for the fixed points of $\hat{S} \circ \hat{h}$. Their proofs are similar to those in the lower dimensional case given in [50]. Given the above lemmas, we can prove convergence of Algorithm 1.

THEOREM 3.4 (convergence). *Under Assumption* 1, *the sequence* $\{(w^k, z^k, u^k)\}$ *generated by Algorithm* 1 *from any starting point* (w^0, z^0, u^0) *converges to a solution* (w^*, z^*, u^*) *of* (1.6).

Proof. The proof is similar to that of Theorem 3.4 in [50] and thus is omitted. □

Next we develop a finite convergence property for the auxiliary variables w and z. Let

$$h_i(v) = \left(h_i^{(1)}(v); \ldots; h_i^{(q)}(v)\right) \in \mathbb{R}^q, \quad i = 1, \ldots, n^2;$$

namely, $h_i(v)$ is the vector formed by stacking the ith components of $h^{(1)}(v), \ldots, h^{(q)}(v)$. We will make use of the following index sets:

$$L_1 = \left\{i, \|G_i u^*\| \equiv \|h_i(v^*)\| < \frac{1}{\beta}\right\}, \quad L_2 = \left\{i, |(Ku^* - f)_i| \equiv \left|h_i^{(q+1)}(v^*)\right| < \frac{\mu}{\beta}\right\},$$

and their complements $E_1 = \{1, \ldots, n^2\} \setminus L_1$ and $E_2 = \{1, \ldots, mn^2\} \setminus L_2$.

THEOREM 3.5 (finite convergence). *Under Assumption* 1, *the sequence* $\{(w^k, z^k, u^k)\}$ *generated by Algorithm* 1 *from any starting point* (w^0, z^0, u^0) *satisfies* $\mathbf{w}_i^k = \mathbf{w}_i^* = 0, \forall\, i \in L_1$, *and* $z_i^k = z_i^* = 0, \forall\, i \in L_2$, *for all but finite numbers of iterations that do not exceed* $\|v^0 - v^*\|^2/\omega_1^2$ *and* $\|v^0 - v^*\|^2/\omega_2^2$, *respectively, where*

$$\omega_1 \triangleq \min_{i \in L_1} \left\{\frac{1}{\beta} - \|h_i(v^*)\|\right\} > 0 \ \text{ and } \ \omega_2 \triangleq \min_{i \in L_2} \left\{\frac{\mu}{\beta} - \left|h_i^{(q+1)}(v^*)\right|\right\} > 0. \tag{3.6}$$

Proof. For any $i \in \{1, \ldots, n^2\}$, it holds that

$$\left\|\mathbf{w}_i^{k+1} - \mathbf{w}_i^*\right\|^2 = \left\|s \circ h_i\left(v^k\right) - s \circ h_i(v^*)\right\|^2 \le \left\|h_i\left(v^k\right) - h_i(v^*)\right\|^2. \tag{3.7}$$

Suppose $\mathbf{w}_i^{k+1} \neq 0$ for some $i \in L_1$, then

$$\begin{aligned}\left\|\mathbf{w}_i^{k+1} - \mathbf{w}_i^*\right\|^2 &= \left\|s \circ h_i\left(v^k\right) - s \circ h_i(v^*)\right\|^2 = \left(\left\|h_i\left(v^k\right)\right\| - 1/\beta\right)^2 \\ &\le \left\{\left\|h_i\left(v^k\right) - h_i(v^*)\right\| - (1/\beta - \|h_i(v^*)\|)\right\}^2 \\ &\le \left\|h_i\left(v^k\right) - h_i(v^*)\right\|^2 - (1/\beta - \|h_i(v^*)\|)^2 \\ &\le \left\|h_i\left(v^k\right) - h_i(v^*)\right\|^2 - \omega_1^2,\end{aligned} \tag{3.8}$$

where the first equality comes from the iteration of $\mathbf{w}_i$ in (2.1) and the definition of $h_i(v)$; the second equality holds because of $\|h_i(v^*)\| < 1/\beta$, $\mathbf{w}_i^{k+1} \neq 0$, and the definition of s; the first inequality is triangular inequality; the second inequality follows from the fact that $\|h_i(v^k) - h_i(v^*)\| \ge 1/\beta - \|h_i(v^*)\| > 0$; and the last one uses the definition of ω_1 in (3.6). Furthermore,

$$\begin{aligned}\left\|z^{k+1} - z^*\right\|^2 &= \left\|s_\mu \circ h^{(q+1)}\left(v^k\right) - s_\mu \circ h^{(q+1)}(v^*)\right\|^2 \\ &\le \left\|h^{(q+1)}\left(v^k\right) - h^{(q+1)}(v^*)\right\|^2.\end{aligned} \tag{3.9}$$

Combining (3.7), (3.8), and (3.9), we get

$$\begin{aligned}(3.10)\quad \left\|v^{k+1}-v^*\right\|^2 &= \left\|w^{k+1}-w^*\right\|^2+\left\|z^{k+1}-z^*\right\|^2\\ &=\sum_{i=1}^{n^2}\left\|\mathbf{w}_i^{k+1}-\mathbf{w}_i^*\right\|^2+\left\|z^{k+1}-z^*\right\|^2\\ &\le \sum_{i=1}^{n^2}\left\|h_i\left(v^k\right)-h_i(v^*)\right\|^2+\left\|h^{(q+1)}\left(v^k\right)-h^{(q+1)}(v^*)\right\|^2-\omega_1^2\\ &=\sum_{j=1}^{q+1}\left\|h^{(j)}\left(v^k\right)-h^{(j)}(v^*)\right\|^2-\omega_1^2=\left\|\hat{h}\left(v^k\right)-\hat{h}(v^*)\right\|^2-\omega_1^2\\ &\le \left\|v^k-v^*\right\|^2-\omega_1^2.\end{aligned}$$

Therefore, for $i\in L_1$, it holds that $\mathbf{w}_i^k=\mathbf{w}_i^*=0$ in no more than $\|v^0-v^*\|^2/\omega_1^2$ iterations.

For any $i\in\{1,\dots,mn^2\}$, we have

$$\begin{aligned}(3.11)\qquad \left(z_i^{k+1}-z_i^*\right)^2&=\left(s_\mu\circ h_i^{(q+1)}\left(v^k\right)-s_\mu\circ h_i^{(q+1)}(v^*)\right)^2\\ &\le\left|h_i^{(q+1)}\left(v^k\right)-h_i^{(q+1)}(v^*)\right|^2.\end{aligned}$$

Similarly, supposing $z_i^{k+1}\neq 0$ for some $i\in L_2$, from $z_i^*=0$ we get

$$\begin{aligned}(3.12)\quad \left(z_i^{k+1}-z_i^*\right)^2&=\left(s_\mu\circ h_i^{(q+1)}\left(v^k\right)\right)^2=\left(\left|h_i^{(q+1)}\left(v^k\right)\right|-\frac{\mu}{\beta}\right)^2\\ &\le\left\{\left|h_i^{(q+1)}\left(v^k\right)-h_i^{(q+1)}(v^*)\right|-\left(\frac{\mu}{\beta}-\left|h_i^{(q+1)}(v^*)\right|\right)\right\}^2\\ &\le\left|h_i^{(q+1)}\left(v^k\right)-h_i^{(q+1)}(v^*)\right|^2-\left(\frac{\mu}{\beta}-\left|h_i^{(q+1)}(v^*)\right|\right)^2\\ &\le\left|h_i^{(q+1)}\left(v^k\right)-h_i^{(q+1)}(v^*)\right|^2-\omega_2^2,\end{aligned}$$

where the reasoning is identical to that of (3.8) and ω_2 is defined in (3.6). Combining (3.7), (3.11), and (3.12), similar to (3.10), we get

$$(3.13)\qquad \left\|v^{k+1}-v^*\right\|^2\le\left\|\hat{h}\left(v^k\right)-\hat{h}(v^*)\right\|^2-\omega_2^2\le\left\|v^k-v^*\right\|^2-\omega_2^2.$$

Therefore, $z_i^k=z_i^*=0$ for $i\in L_2$ in no more than $\|v^0-v^*\|^2/\omega_2^2$ iterations. ∎

Given the finite convergence of $\mathbf{w}_i^k=\mathbf{w}_i^*=0$ for $i\in L_1$ and $z_i^k=z_i^*=0$ for $i\in L_2$, we next show the q-linear convergence of u^k and the remaining components in v^k. For convenience, let

$$L=L_1\cup\left(n^2+L_1\right)\cup\ldots\cup\left((q-1)n^2+L_1\right)\cup\left(qn^2+L_2\right)$$

and $E=\{1,\dots,(q+m)n^2\}\setminus L$ be the complement of L. Let v_L be the subvector of v with components $\{v_i : i\in L\}$ and v_E be defined similarly. Furthermore, let $P=HM^{-1}H^\top$ and $P_{EE}=[P_{i,j}]_{i,j\in E}$. From the definition of M, it is obvious that P is a projection matrix and thus $P^2=P$.

THEOREM 3.6 (q-linear convergence). *Under Assumption* 1, *the sequence* $\{(v^k,u^k)=(w^k,z^k,u^k)\}$, *generated by Algorithm* 1, *satisfies*

1. $\|v_E^{k+1} - v_E^*\| \le \sqrt{\rho(P_{EE})}\|v_E^k - v_E^*\|$;
2. $\|u^{k+1} - u^*\|_M \le \sqrt{\rho(P_{EE})}\|u^k - u^*\|_M$,

for all sufficiently large k.

Proof. From (3.2)–(3.5) and the nonexpansiveness of S and s_μ, we get

$$u^{k+1} - u^* = M^{-1}H^\top \left(v^{k+1} - v^*\right) \tag{3.14}$$

and

$$\begin{aligned} \left\|v^{k+1} - v^*\right\|^2 &= \left\|w^{k+1} - w^*\right\|^2 + \left\|z^{k+1} - z^*\right\|^2 \\ &= \left\|S\left(G^{(1)}u^k; \dots ; G^{(q)}u^k\right) - S\left(G^{(1)}u^*; \dots ; G^{(q)}u^*\right)\right\|^2 \\ &\quad + \left\|s_\mu\left(Ku^k - f\right) - s_\mu\left(Ku^* - f\right)\right\|^2 \\ &\le \left\|G\left(u^k - u^*\right)\right\|^2 + \left\|K\left(u^k - u^*\right)\right\|^2 = \left\|H\left(u^k - u^*\right)\right\|^2. \end{aligned} \tag{3.15}$$

Combining the recursion (3.14), (3.15), and the definition of P, it holds that

$$\left\|v^{k+1} - v^*\right\|^2 \le \left\|HM^{-1}H^\top\left(v^k - v^*\right)\right\|^2 = \left\|P\left(v^k - v^*\right)\right\|^2.$$

Since we are interested only in the asymptotic behavior of Algorithm 1, without loss of generality, we assume that $v_L^k = v_L^* = 0$. Further from $P^2 = P$, the above inequality becomes

$$\left\|v_E^{k+1} - v_E^*\right\|^2 \le \left(v_E^k - v_E^*\right)^\top P_{EE}\left(v_E^k - v_E^*\right) \le \rho(P_{EE})\left\|v_E^k - v_E^*\right\|^2,$$

which implies Assertion 1 of this theorem. Multiplying H on both sides of (3.14), from $v_L^k = 0$ and (3.15), we get

$$\left\|H\left(u^{k+1} - u^*\right)\right\|^2 \le \rho(P_{EE})\left\|v^{k+1} - v^*\right\|^2 \le \rho(P_{EE})\left\|H\left(u^k - u^*\right)\right\|^2.$$

Recall that $M = H^\top H$. The above inequality implies assertion 2 of this theorem. ∎

Theorem 3.6 states that Algorithm 1 generates a sequence of points that converge q-linearly with a convergence rate depending on the spectral radius of the submatrix P_{EE} rather than that of the whole matrix. Since P is a projection matrix and P_{EE} is a minor of P, it holds that $\rho(P_{EE}) \le \rho(P) = 1$.

4. Numerical results. In this section, we present numerical results of recovering images by the proposed alternating minimization algorithm. In our experiments, we used two images, grayscale image Cameraman (256×256) and RGB color image Rose (303×250) introduced in [7], with different blurs and noise. Two types of impulsive noise were used in the test: the salt-and-pepper noise and the random-valued impulsive noise. In the rest of this section, we first describe the test platform and a practical implementation of Algorithm 1, then compare our algorithm with the algorithms in [26] and [9] for grayscale image deblurring. We next present color image results, and finally summarize the performance of our algorithm.

4.1. Test platform and practical implementation. We implemented Algorithm 1 in MATLAB and generated all blurring effects using the MATLAB function "`imfilter`" with periodic boundary conditions. The experiments were performed under Windows Vista Premium and MATLAB v7.6 (R2008a) running on a Lenovo laptop with an Intel Core 2 Duo CPU at 1.8 GHz and with 2 GB of memory.

Let $\mathbf{E}(\bar{u})$ be the mean intensity value of the original image $\bar{u}$, and let u be the restored image. We measure the quality of restoration by signal-to-noise ratio (SNR), which is defined as

$$\text{SNR} \triangleq 10 * \log_{10} \frac{\|\bar{u} - \mathbf{E}(\bar{u})\|^2}{\|\bar{u} - u\|^2}.$$

Although Algorithm 1 is applicable to several types of regularization terms based on TV, weighted TV, and high-order derivatives (see [51] for details), we limited our experiments on model (1.3), i.e., $G_i = I_m \otimes D_i$, and $\alpha_i \equiv 1$ in (1.4). In our implementation, we set $\beta_1 = 2^{10}$ and $\beta_2 = 2^{15}\mu$ in (1.6), while the value of μ is determined experimentally. Based on our experimental results, these β-values are large enough for the algorithm to attain an SNR value that can hardly be further improved by increasing the penalty parameters.

From (2.1) and (2.2), the smaller the penalty parameters are, the more zeros the shrinkage will produce. To speed up convergence, we implemented a continuation scheme on the penalty parameters; that is, let β_1 and β_2 (or γ after dividing μ) take small values at the beginning and gradually increase them to the prescribed values. Specifically, we tested the γ-sequence $2^0, 2^1, 2^2 \ldots, 2^{15}$. Accordingly, β_1 was set to $2^0, 2^{2/3}, 2^{4/3} \ldots, 2^{10}$. Continuation techniques are widely used with penalty methods and, for our problem, its use is also theoretically well-justified by Theorem 3.6. From the definitions of L and E, it is likely that smaller penalty parameters yield smaller E and thus fast convergence. As such, earlier subproblems with smaller penalty parameters can be solved quickly, and the later subproblems can also be solved relatively quickly with warm starts from previous solutions.

To sum up, our practical implementation of Algorithm 1 involves two loops. The outer loop increases β_1 and γ from 1 to 2^{10} and 2^{15}, respectively. For fixed β_1 and γ, the inner loop solves (1.6) until (2.16) is met. Although the above framework can be modified with much flexibility, e.g., adaptively increasing the penalty parameters and selecting ϵ from one outer iteration to another, this basic implementation already works quite well. Following [50], we give the name *fast total variation deconvolution*, or FTVd, to Algorithm 1 with the prescribed continuation scheme.

4.2. Comparison with the linear program approach in [26]. In this subsection, we compare the performance of FTVd with that of the algorithm in [26], where the authors converted the deblurring model

$$\min_u \left\{ \sum_i \|D_i u\|_1 + \mu \|Ku - f\|_1 : u \geq 0 \right\} \tag{4.1}$$

into a linear program and proposed to solve it by a primal-dual interior point method. Following their naming, we refer to both the algorithm in [26] and solution of (4.1) as the least absolute deviation or LAD. Their approach requires solving a positive definite linear system at each iteration by a preconditioned conjugate gradient (PCG) method with a sparse inverse preconditioner [4, 33, 46]. Specially, they used the factorized banded inverse preconditioner (FBIP, cf. [34]), which has a triangular block banded structure with each block being also banded. Let p be the block-level lower bandwidths of the preconditioner, and let q be the (lower or upper) bandwidths of each block. As the authors pointed out, when the kernel size is large, it is necessary to use large (p, q) to obtain a sufficiently good preconditioner. However, the computational cost of FBIP at each outer iteration is $O(p^3 q^3 n^2)$ that increases quickly with p and q.

Therefore, for large-sized blurring kernels, a balance needs to be found between the PCG iteration numbers and the cost of preconditioner computation. Since the linear systems involved are increasingly ill conditioned towards the end of outer iterations, a reasonable implementation of their algorithm should be to use the FBIP only when the plain CG method encounters difficulties. For fairness of comparison, we utilized a diagonal preconditioner (similar to the one used in [43] that is included in the LAD code provided to us by the authors of [26]) at the beginning and resorted to the FBIP only when the iteration number required by the MATLAB `pcg` code exceeded 100 to reach a relative residue of 10^{-5}. This way, the wasteful cost of computing the FBIP at the early iterations was avoided.

Since the LAD code used in our experiments is applicable only to grayscale images, we used Cameraman for comparison. Furthermore, since the LAD algorithm solves only the 1-norm based, anisotropic TVL1 problem, we used our algorithm to solve (1.3) with $\|\cdot\|_2$ being replaced by $\|\cdot\|_1$. This modification requires changing (2.1) to

$$\mathbf{w}_i = \max\{|G_i u| - \alpha_i/\beta, 0\} \circ \operatorname{sgn}(G_i u),\ i = 1, \ldots, n^2,$$

where $|\cdot|$ represents componentwise absolute value. The optimality conditions (2.11)–(2.13) and the stopping criterion (2.16) were modified accordingly. There are also two differences in the problems solved by these two algorithms. First, problem (4.1) enforces nonnegativity on u while FTVd does not deal with this requirement. This difference affects the solutions, but only to a small extent. Without the nonnegativity constraints in (4.1), LAD does not run faster because it would need to split u into u^+, u^- and require $u = u^+ - u^-, u^+, u^- \geq 0$. The differences of the two algorithms in terms of CPU time and restoration quality are given in the next two paragraphs. Second, the LAD algorithm uses the Neumann boundary conditions instead of the periodic boundary conditions that FTVd uses. The influence of boundary conditions on image quality was also negligible because the sizes of the tested images are much larger than that of the tested blurring kernels. For more details about boundary conditions, see [36].

Based on the above discussions, we set $p = 4$ and $q = 7$ in LAD and stopped LAD once the normalized duality gap was less than 5×10^{-4}. In this experiment, we applied the Gaussian blur of the size 7×7 and standard deviation 5. For LAD, after we generated the blurry image by MATLAB function "`imfilter`" with symmetric boundary conditions, we corrupted 40% to 80% of pixels of the blurry image at random with salt-and-pepper noise. For FTVd, we first generated the blurry image with periodic boundary conditions and then corrupted the blurry image by exactly the same salt-and-pepper noise recorded when generating noise for LAD. The original blurry and noisy Cameraman images and their restorations by LAD and FTVd are given in Figure 4.1, where the values of μ, CPU time, and SNRs of the restorations are also given. Since the blurry and noisy images of LAD and FTVd have no visible difference, we plotted only the blurry and noisy images for FTVd in Figure 4.1.

As can be seen from Figure 4.1, the results of FTVd and LAD have similar quality in all the test cases as the two methods essentially solve the same TVL1 model (besides the nonnegativity used in LAD). Clearly, the restoration quality of both deteriorated as the noise level increased, and both were unable to produce acceptable restoration quality at the 80% noise level. Since the quality of restoration is ultimately determined by the underlying TVL1 model, these experiments have reaffirmed an intrinsic limitation of the TVL1 model; that is, the model would work poorly once the percentage of the corrupted pixels reached a certain high level.

FIG. 4.1. *First column from top to bottom: original Cameraman image, blurred image by Gaussian kernel of size* 7×7 *and standard deviation* 5 *with the Neumann boundary conditions (SNR:* 9.69*dB), blurred image by the same kernel with periodic boundary conditions (SNR:* 9.57*dB), respectively; second to fourth column, top row: the blurry image is corrupted by* 40%*,* 60%*, and* 80% *salt-and-pepper noise, respectively; second to fourth column, middle row: results recovered by LAD method (CPU time:* 491*,* 493*,* 905 *seconds; SNRs:* 14.47*,* 11.78*,* 8.08*dB); second to fourth column, bottom row: results recovered by FTVd (*$\epsilon \equiv 0.001$ *in* (2.16)*;* $\mu = 36, 10, 2$*; CPU time:* 31*,* 30*,* 32 *seconds; SNRs:* 14.81*,* 11.62*,* 8.09*dB, respectively).*

By comparing the CPU times used by the two methods, we conclude that FTVd is much faster than LAD. In these tests, we set $p = 4$ and $q = 7$ in LAD because we tried with a great deal of effort and failed to run LAD to the prescribed accuracy with smaller p and q due to numerical singularity in FBIP that caused the MATLAB `pcg` code to exit without producing an approximate solution. When the kernel size became larger, the linear system that LAD needed to solve became even more ill-conditioned. As a result, the diagonal preconditioners used in [26] were not sufficient to greatly improve the ill-conditioning of the linear systems, and denser FBIPs became essential. As pointed out before, the computation of each FBIP was $O(p^3q^3n^2)$, which increases quickly with p and q. In comparison, the performance of FTVd is not affected by the increase of the kernel size because it avoids solving any linear systems iteratively. To illustrate this, we applied the Gaussian kernel of the size 15×15 and standard deviation 9 to the Cameraman image and then corrupted the blurry image by 60% salt-and-pepper noise. The blurry image, blurry and noisy image, and recovered result by FTVd are shown in Figure 4.2, from which we see that the CPU time did not increase. However, we were unable to solve (4.1) to the prescribed accuracy by setting $p = 4$ and $q = 7$ in LAD for this blur. We tried larger p and q values without success because for larger (p, q) LAD simply took too much memory and CPU time to run on the aforementioned laptop computer.

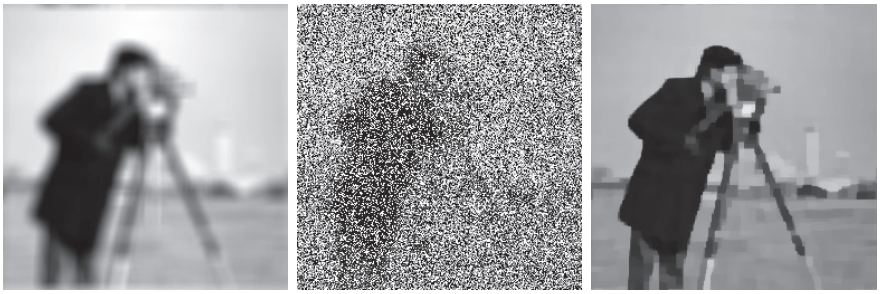

FIG. 4.2. *Left: Cameraman image blurred by Gaussian kernel of size* 15×15 *and standard deviation* 9*, SNR:* 7.23*dB; middle: the blurred image is corrupted by* 60*% salt-and-pepper noise; right: result recovered by FTVd (*$\epsilon = 0.001$ *in* (2.16)*;* $\mu = 10$*, CPU time:* 26 *seconds, SNR:* 10.38*dB).*

4.3. Comparison with the two-phase method in [9]. In this subsection, we compare FTVd with the two-phase method recently proposed in [9]. In the two-phase method, locations of impulsive noise are detected using certain median-type filters in the first phase. Then, the original image is recovered through a variational framework in the second phase where only a part of the data, estimated as noise-free, is used. Specifically, the authors of [9] use the adaptive median filter [30] to detect salt-and-pepper noise and the adaptive center-weighted median filter [32] to detect random-valued impulse noise.

Let $\Omega \subset \mathbb{R}^2$ be a region on which the original image $\bar{u}$ is defined and $\mathcal{U} \subset \Omega$ be a subset on which the observed data have been estimated to be noise-free. In the second phase, the two-phase method approximately solves the following variational model:

$$\min_{u,\Gamma} \beta \int_{\Omega\setminus\Gamma} \|\nabla u\|^2 \mathrm{d}x + \alpha \int_{\Gamma} \mathrm{d}\sigma + \int_{\mathcal{U}} |\kappa * u - f| \mathrm{d}x, \tag{4.2}$$

where $\alpha, \beta > 0$ are regularization parameters, and $\Gamma \subset \Omega$ represents edges in an image. The first two terms in the objective function of (4.2) are known as the Mamford–Shah (MS) regularizer [35] and the last term fits the noise-free data. Since MS regularization treats smooth regions and object contours separately, it is known to preserve edges well. In the literature, the MS regularizer is usually approximated by a sequence of functionals in the framework of Γ-convergence (see, for example, [20, 24]). In [9], the authors approximated (4.2) by

$$\min_{u,v} \beta \int_{\Omega} v^2 \|\nabla u\|^2 \mathrm{d}x + \alpha \int_{\Omega} \left(\epsilon \|\nabla v\|^2 + \frac{(v-1)^2}{4\epsilon} \right) \mathrm{d}x + \int_{\mathcal{U}} \sqrt{(\kappa * u - f)^2 + \eta} \mathrm{d}x, \tag{4.3}$$

where $\epsilon, \eta > 0$ are small parameters and $v : \Omega \to \mathbb{R}$ represents the contours of u. Then a solution to (4.3) was obtained via alternately solving the two equations in the Euler–Lagrange system of (4.3), while one of the two variables is fixed. We note that the two-phase method can handle impulsive plus Gaussian noise by incorporating another fidelity term in the second phase.

Since the model of the two-phase method in [9] is significantly different from the pure variation model (1.3) that we solve, we shall compare both restoration quality and speed using the Cameraman image. In the comparison, the image was blurred by

FIG. 4.3. *First column from top to bottom: original Cameraman image, blurred image by out-of-focus kernel of radius* 7 *with the Neumann boundary conditions (SNR:* 7.77*dB), blurred image by the same kernel with periodic boundary conditions (SNR:* 7.64*dB), respectively; second to fourth column, top row: the blurry image is corrupted by* 25%, 40%, 55% *random-valued noise, respectively; second to fourth column, middle row: results recovered by the two-phase method (parameters in* (4.3)*:* $\eta \equiv 10^{-4}$, $(\alpha, \beta, \epsilon) = (1, 1, 0.5) \times 10^{-3}, (2, 2, 0.5) \times 10^{-3}, (5, 5, 0.1) \times 10^{-3}$, *SNRs:* 15.46, 14.04, 10.41*dB, CPU:* 1315, 1434, 1396 *seconds, respectively); second to fourth column, bottom row: results recovered by FTVd (*$\epsilon = 0.001$ *in* (2.16), $\mu : 150, 45, 10$, *SNRs:* 18.17, 14.00, 9.33*dB, CPU time:* 54, 47, 41 *seconds, respectively).*

an out-of-focus kernel of radius 7 and then corrupted by 25%, 40%, and 55% random-valued impulsive noise. In the code from the authors of [9], Neumann boundary conditions are used while FTVd uses periodic boundary conditions. The original blurred images with two types of boundary conditions and the recovered results by both methods are given in Figure 4.3. Since the difference in boundary conditions does not generate visible differences in resulting blurry and noisy images, we only give the ones corresponding to the periodic boundary conditions in the first row of Figure 4.3. For the two-phase method, we used the exactly same parameter values as those used in [9]. These parameter values are given in the caption of Figure 4.3 (see the expression in (4.3) for the parameters involved), along with parameter values used by FTVd. The CPU time used by both methods and SNRs of the restorations is also given in the caption of Figure 4.3.

As can be seen from Figure 4.3, FTVd gives better restoration quality when the corruption of impulsive noise is relatively low at 25%; the two-phase method gives much better quality when the noise level becomes high at 55%, while their performances, qualitywise, are about even when the noise is at the 40% level. On the other hand, in terms of restoration speed, FTVd is always faster, by more than

one order of magnitude on this 256 by 256 image, than the two-phase method as implemented in [9].

It should be intuitively clear that the robustness of the two-phase method with respect to high level impulsive noise is the result of detecting and removing corrupted pixels done in the first phase. The exclusion of these corrupted pixels from the second phase—the restoration phase—makes it possible to produce cleaner restored images. On the other hand, the TVL1 model (1.3) includes all corrupted pixels, which can greatly affect the restoration process when the percentage of corrupted data is too high.

The fast speed of FTVd comes mainly from its low-complexity iterations, consisting of shrinkage, FFTs, and finite differences. The method also has a reasonable convergence rate. On the other hand, the two-phase method requires solving large and ill-conditioned linear systems of equations at each iteration.

We also compared the two methods with different image sizes and blurring kernels. As the image size increases, the speed of the two-phase method (as implemented in [9]) slows down quickly, while FTVd slows down at a much more moderate rate. Furthermore, the speed of FTVd is not affected by the increase of the blurring kernel size, while larger kernel sizes increase the degree of ill conditioning of the linear systems solved by the two-phase method.

4.4. Color image results. In this subsection, we present recovery results for color images by FTVd. We first blurred the color image of the rose by cross-channel blurring described below and then corrupted 40% to 60% of its pixels at random by random-valued noise. Let (A, `hsize`) denote the average blur of the size `hsize`, (G, `hsize`, `sigma`) the Gaussian blur of the size `hsize` and standard deviation `sigma`, and (M, `len`, `theta`) the motion blur with motion length `len` and angle `theta`. We chose a "diagonally dominant" cross-channel blurring kernel:

$$\text{(4.4)}\qquad \begin{bmatrix} H_{rr} & H_{rg} & H_{rb} \\ H_{gr} & H_{gg} & H_{gb} \\ H_{br} & H_{bg} & H_{bb} \end{bmatrix} = \begin{bmatrix} 0.8\cdot(A,9) & 0.1\cdot(A,9) & 0.1\cdot(A,9) \\ 0.15\cdot(G,11,5) & 0.7\cdot(G,11,5) & 0.15\cdot(G,11,5) \\ 0.2\cdot(M,21,135) & 0.2\cdot(M,21,135) & 0.6\cdot(M,21,135) \end{bmatrix},$$

where $H_{\sigma_1\sigma_2}$ defines within-channel blurring for $\sigma_1 = \sigma_2$ and cross-channel blurring for $\sigma_1 \neq \sigma_2$. Considering that within-channel blurs are usually stronger than cross-channel ones, we assigned larger weights to the within-channel blurs. Similar methods for choosing kernel weights are used in the literature; see, e.g., [25, 27]. We note that the types, locations, and kernel size appear to have little influence on the efficiency of FTVd. Furthermore, our algorithm converges well as long as Assumption 1 is not violated; see [51] for more discussions. The blurry and noisy images and their restorations from FTVd are given in Figure 4.4 along with the values of μ, CPU time, and SNRs.

Generally, it is more difficult to remove random-valued noise than salt-and-pepper noise because the former has a wider range of intensity values. As can be seen from Figure 4.4, the restored images are cleaner when the amount or random noise is smaller. The required CPU times are significantly longer than those reported for the grayscale image in Figures 4.1, 4.2, and 4.3 because the per-iteration computational cost has increased significantly from deblurring single-channel images to multichannel ones.

FIG. 4.4. *First column: on the top is the original rose image and on the bottom is the blurred image by the kernel given in* (4.4) *(SNR:* 8.15*dB). Second to fourth column: the blurry image is corrupted by* 40*%,* 50*%, and* 60*% random-valued noise (top row); results recovered by FTVd (bottom row from left to right:* $\epsilon \equiv 0.005$ *in* (2.16)*;* $\mu = 8, 4, 2$*; CPU time:* 177*,* 226*, and* 209 *seconds; SNRs:* 16.43*,* 14.36*, and* 10.56 *dB, respectively).*

4.5. Summary. In the proposed Algorithm 1, there are two steps at each iteration. The first step computes $\mathbf{w}$ and z by shrinkages ((2.1) and (2.2)), and the second solves (2.5) for u by FFTs and block Gaussian elimination (for multichannel images). The shrinkage operations in the first step have a linear complexity in terms of n^2. Therefore, the main computational work lies in the second step. For grayscale images where $m = 1$, a total number of 2 FFTs (including 1 inverse FFT) are needed to solve (2.5), while no Gaussian elimination is necessary. For color images where $m = 3$, a total number of 6 FFTs (including 3 inverse FFTs) are needed to solve (2.5), plus about $18n^2$ arithmetic operations are required by the block Gaussian elimination. If higher-order derivatives are used in regularization, the required number of finite differences would increase, but not that of FFTs. Upon profiling our code on restoring color images, we observed that about 40% of the total CPU time was spent on Gaussian elimination and checking stopping conditions, while the rest was spent on FFTs and other calculations. The CPU time of FTVd changes little when the kernel size varies and increases at a moderate rate as the image size increases.

In summary, our numerical results have shown that when noise level is high, the two-phase method can deliver better restoration quality than FTVd can, while in all cases FTVd has a much faster restoration speed.

5. Concluding remarks. An alternating minimization algorithm is proposed for solving the TVL1-like problem (1.4). The algorithm is applicable to both the isotropic and anisotropic TV discretizations, and has finite convergence for some auxiliary variables and a q-linear convergence rate for the rest. At each iteration, the total computational cost is dominated by the costs of a number of FFTs and a block Gaussian elimination. Our numerical results show that the algorithm is efficient and stable for solving the TVL1 model (1.3). For grayscale images and a relatively small

kernel size, our algorithm is already over an order of magnitude faster than the LAD algorithm (as implemented in [26]), previously regarded as an efficient algorithm, and the advantage further widens as the kernel size increases.

Our numerical results show that during deblurring, the algorithm is capable of removing a considerable amount of impulsive noise that corrupts up to 60% of the pixels. However, with higher levels of impulsive noise, the underlying TVL1 model (1.3) is unable to produce high quality restorations due to the fact that all corrupted pixels are included in the fidelity term of the model (1.3). This scheme of including all data allows us to efficiently solve (2.5) by FFTs, but unfortunately also puts a limit on the percentage of corrupted pixels that can be effectively removed. On the other hand, removing corrupted pixels from a restoration model, as is done in the two-phase method, can enhance restoration quality, but the resulting coefficient matrices lose the block circulant structure so that fast transforms cannot be directly applied to the involved linear systems.

The splitting and alternating minimization approaches used in this paper have been successfully applied to a number of different models, such as the TVL2 model [50] and its multichannel extension [51]. More recently, this technique has also been extended to TV-based image reconstruction from incomplete Fourier data [52] that may find applications in magnetic resonance imaging. In all these cases, the resulting algorithms achieve their good performance from the stability and efficiency of shrinkage and FFT that constitute the main computations of these algorithms. The development of a fast algorithm for deconvolution that is built on shrinkage and FFT and can handle partial observation data (excluding corrupted data as in the second stage of the two-phase method in [9]) is certainly a useful topic for further research.

Acknowledgments. The authors would like to thank Dr. Haoying Fu and Dr. Jianfeng Cai for providing the code of [26] and that of [9], respectively. The authors are grateful to three anonymous referees for their constructive suggestions and valuable comments that have helped improve the paper greatly.

REFERENCES

[1] S. Alliney, *Digital filters as absolute norm regularizers*, IEEE Trans. Signal Process., 40 (1992), pp. 1548–1562.
[2] J. Astola and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*, CRC, Boca Raton, FL, 1997.
[3] L. Bar, A. Brook, N. Sochen, and N. Kiryati, *Deblurring of color images corrupted by impulsive noise*, IEEE Trans. Image Process., 16 (2007), pp. 1101–1110.
[4] M. Benzi, C. Meyer, and M. Tuma, *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput., 17 (1996), pp. 1135–1149.
[5] P. Blomgren and T. F. Chan, *Color TV: Total variation methods for restoration of vector-valued images*, IEEE Trans. Image Process., 7 (1998), pp. 304–309.
[6] A. Bovik, *Handbook of Image and Video Processing*, Academic, New York, 2000.
[7] X. Bresson and T. F. Chan, *Fast minimization of the vectorial total variation norm and applications to color image processing*, UCLA CAM Report 07–25, 2007.
[8] A. Brook, R. Kimmel, and N. Sochen, *Variational restoration and edge detection for color images*, J. Math. Imaging Vision, 18 (2003), pp. 247–268.
[9] J. F. Cai, R. Chan, and M. Nikolova, *Two phase methods for deblurring images corrupted by impulse plus Gaussian noise*, AIMS J. Inverse Prob. Imaging, 2 (2008), pp. 187–204.
[10] T. F. Chan and S. Esedoglu, *Aspects of total variation regularized L1 function approximation*, SIAM J. Appl. Math., 65 (2005), pp. 1817–1837.
[11] R. Chan, C. W. Ho, and M. Nikolova, *Salt-and-pepper noise removal by median-type noise detector and detail-preserving regularization*, IEEE Trans. Image Process., 14 (2005), pp. 1479–1485.

[12] R. Chan, C. Hu, and M. Nikolova, *An iterative procedure for removing random-valued impulse noise*, IEEE Signal Process. Lett., 11 (2004), pp. 921–924.
[13] T. F. Chan, S. H. Kang, and J. Shen, *Total variation denoising and enhancement color images based on the CB and HSV color models*, J. Visual Commun. Image Rep., 12 (2001), pp. 422–435.
[14] T. F. Chan and J. Shen, *Variational restoration of non-flat image features: Models and algorithms*, SIAM J. Appl. Math., 61 (2000), pp. 1338–1361.
[15] T. Chen, T. Huang, W. Yin, and X. S. Zhou, *A new coarse-to-fine framework for 3D brain MR image registration*, in Computer Vision for Biomedical Image, Lecture Notes in Computer Science 3765, Springer, 2005, New York, pp. 114–124.
[16] T. Chen, W. Yin, X. S. Zhou, D. Comaniciu, and T. Huang, *Total variation models for variable lighting face recognition*, IEEE Trans. Pattern Anal. Machine Intelligence (PAMI), 28 (2006), pp. 1519–1524.
[17] T. Chen, W. Yin, X. S. Zhou, D. Domaniciu, and T. Huang, *Illumination normalization for face recognition and uneven background correction using total variation based image models*, in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 2, San Diego, CA, 2005, pp. 532–539.
[18] T. Chen and H. R. Wu, *Space variant median filters for the restoration of impulse noise corrupted images*, IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., 48 (2001), pp. 784–789.
[19] R. Courant, *Variational methods for the solution of problems with equilibrium and vibration*, Bull. Amer. Math. Soc., 49 (1943), pp. 1–23.
[20] G. Dal Maso, *An introduction to* Γ*-convergence*, in Progress in Nonlinear Differential Equations and their Applications, Birkhäuser, 1993.
[21] J. Darbon and M. Sigelle, *Image restoration with discrete constrained total variation Part* I: *Fast and exact optimization*, J. Math. Imaging Vision, 26 (2006), pp. 261–276.
[22] D. Dobson and F. Santosa, *Recovery of blocky images from noisy and blurred data*, SIAM J. Appl. Math., 56 (1996), pp. 1181–1198.
[23] H. L. Eng and K. K. Ma, *Noise adaptive soft-switching median filter*, IEEE Trans. Image Process., 10 (2001), pp. 242–251.
[24] S. Esedoglu and J. Shen, *Digital inpainting based on the Mumford-Shah-Euler image model*, European J. Appl. Math., 13 (2002), pp. 353–370.
[25] H. Y. Fu, M. K. Ng, and J. L. Barlow, *Structured total least squares for color image restoration*, SIAM J. Sci. Comput., 28 (2006), pp. 1100–1119.
[26] H. Y. Fu, M. K. Ng, M. Nikolova, and J. L. Barlow, *Efficient minimization methods of mixed l2-l1 and l1-l1 norms for image restoration*, SIAM J. Sci. Comput., 27 (2006), pp. 1881–1902.
[27] N. P. Galatsanos, A. K. Katsaggelos, R. T. Chan, and A. D. Hillery, *Least squares restorations of multichannel images*, IEEE Trans. Signal Process., 39 (1991), pp. 2222–2236.
[28] D. Geman and G. Reynolds, *Constrained restoration and the recovery of discontinuities*, IEEE Trans. Pattern Anal. Machine Intelligence, 14 (1992), pp. 367–383.
[29] D. Geman and C. Yang, *Nonlinear image recovery with half-quadratic regularization*, IEEE Trans. Image Process., 4 (1995), pp. 932–946.
[30] H. Hwang and R. A. Haddad, *Adaptive median filters: New algorithms and results*, IEEE Trans. Image Process., 4 (1995), pp. 499–502.
[31] R. Kaftory, N. Sochen, and Y. Zeevi, *Color image denoising and blind deconvolution using the Beltrami operator*, in Proceedings of the 3rd International Symposium on ISPA, 1 (2003), pp. 1–4.
[32] S.-J. Ko and Y. H. Lee, *Center weighted median filters and their applications to image enhancement*, IEEE Trans. Circuits Syst., 38 (1991), pp. 984–993.
[33] L. Kolotilina and A. Yeremin, *Factorized sparse approximate inverse preconditionings* I. *Theory*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 45–58.
[34] F.-R. Lin, M. K. Ng, and W.-K. Ching, *Factorized banded inverse preconditioners for matrices with Toeplitz structure*, SIAM J. Sci. Comput., 26 (2005), pp. 1852–1870.
[35] D. Mumford and J. Shah, *Optimal approximations by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42 (1989), pp. 577–685.
[36] M. K. Ng, R. H. Chan, and W. Tang, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–866.
[37] M. Nikolova, *Local strong homogeneity of a regularized estimator*, SIAM J. Appl. Math., 61 (2000), pp. 633–658.
[38] M. Nikolova, *Minimizers of cost-functions involving non-smooth data fidelity terms. Application to the processing of outliers*, SIAM J. Numer. Anal., 40 (2002), pp. 965–994.

[39] M. Nikolova, *A variational approach to remove outliers and impulse noise*, J. Math. Imaging Vision, 20 (2004), pp. 99–120.
[40] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 1999.
[41] G. Pok, J. C. Liu, and A. S. Nair, *Selective removal of impulse noise based on homogeneity level information*, IEEE Trans. Image Process., 12 (2003), pp. 85–92.
[42] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[43] M. Saunders, *Interior methods for optimization with application to maximum entropy problems*, http://sccm.stanford.edu/csri/Saunders.pdf.
[44] O. Scherzer, W. Yin, and S. Osher, *Slope and G-set characterization of set-valued functions and applications to non-differentiable optimization problems*, Commun. Math. Sci., 3 (2005), pp. 479–492.
[45] N. Sochen, R. Kimmel, and R. Malladi, *A general framework for low level vision*, IEEE Trans. Image Process., 7 (1998), pp. 310–318.
[46] W. Tang, *Toward an effective sparse approximate inverse preconditioner*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 970–986.
[47] B. Tang, G. Sapiro, and V. Caselles, *Color image enhancement via chromaticity diffusion*, Tech. Report, Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, 1999.
[48] A. Tikhonov and V. Arsenin, *Solution of Ill-Posed Problems*, Winston, Washington, DC, 1977.
[49] C. R. Vogel and M. E. Oman, *Fast, robust total variation based reconstruction of noisy, blurred images*, IEEE Trans. Image Process., 7 (1998), pp. 813–824.
[50] Y. Wang, J. Yang, W. Yin, and Y. Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imag. Sci., 1 (2008), pp. 248–272.
[51] J. Yang, W. Yin, Y. Zhang, and Y. Wang, *A fast algorithm for edge-preserving variational multichannel image restoration*, SIAM J. Imag. Sci., 2 (2009), pp. 569–592.
[52] J. Yang, Y. Zhang, and W. Yin, *A fast TVL1-L2 minimization algorithm for signal reconstruction from partial fourier data*, Technical report 08-27, CAAM, Rice University, Houston, TX, submitted to J-STSP.
[53] W. Yin, T. Chen, X. S. Zhou, and A. Chakraborty, *Background correction for cDNA microarray image using the $TV + L^1$ model*, Bioinformatics, 21 (2005), pp. 2410–2416.
[54] W. Yin, D. Goldfarb, and S. Osher, *Image cartoon-texture decomposition and feature selection using the total variation regularized L^1 functional*, in Variational, Geometric, and Level Set Methods in Computer Vision, Leture Notes in Computer Science 3752, Springer, New York, 2005, pp. 73–84.
[55] W. Yin, D. Goldfarb, and S. Osher, *The total variation regularized L^1 model for multiscale decomposition*, SIAM J. Multiscale Model. Simul., 6 (2006), pp. 190–211.