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 [38, 39] 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 (TV), either isotropic or anisotropic, and a data fidelity term measured in the L_1-norm. 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 global q-linear convergence. 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 an algorithm recently proposed in [20] that uses a linear program and an interior point method for recovering grayscale images.

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

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

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 $\bar{u} = [\bar{u}^{(1)}; \ldots; \bar{u}^{(m)}] \in \mathbb{R}^{mn^2}$, where $\bar{u}^{(j)} \in \mathbb{R}^{n^2}$ represents the jth channel, $j = 1, \ldots, m$. The observation $f \in \mathbb{R}^{mn^2}$ of $\bar{u}$ is

$$f = K\bar{u} + \omega, \tag{1.1}$$

where

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

is the cross-channel blurring operator, and $\omega \in \mathbb{R}^{mn^2}$ represents additive noise. 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 TV regularization problem with a 1-norm data fidelity term:

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

[*] Department of Mathematics, Nanjing University, 22 Hankou Road, Nanjing, Jiangsu Province, 210093, P.R. China (jfyang2992@yahoo.com.cn)

[†] Department of Computational and Applied Mathematics, Rice University, 6100 Main Street, MS-134, Houston, Texas, 77005, U.S.A. (yin.zhang, wotao.yin@rice.edu)

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 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 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 any pixel i and for some positive integer q, $G_i \in \mathbb{R}^{q \times mn^2}$ is a general local finite difference operator, $\alpha_i > 0$ is a weighting parameter, and $\mu > 0$ balances the regularization term and the fidelity term. 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 either from the classic quadratic penalty function technique in optimization, dated back to Courant's work [17] in 1943, or from the half-quadratic technique initially proposed by Geman and Yang in [22]. In this paper, our derivation follows the former technique for its simplicity. A derivation based on the half-quadratic technique can be found in [39] for a deblurring model under Gaussian noise.

Let $z \in \mathbb{R}^{mn^2}$ and $\mathbf{w}_i \in \mathbb{R}^q$, $i = 1, \ldots, n^2$, be auxiliary variables that approximate $Ku - f$ and $G_i u$ in the nondifferentiable norms in (1.4), respectively. Then, by adding quadratic terms to penalize the difference between every pair of original and auxiliary variables, we obtain the following approximate problem to (1.4):

$$\min_{\mathbf{w},z,u} \sum_{i=1}^{n^2} \left(\alpha_i \|\mathbf{w}_i\|_2 + \frac{\beta}{2} \|\mathbf{w}_i - G_i u\|_2^2 \right) + \mu \left(\|z\|_1 + \frac{\gamma}{2} \|z - (Ku - f)\|_2^2 \right), \tag{1.5}$$

where $\beta, \gamma \gg 0$ are penalty parameters. Let $\mathbf{w} = [\mathbf{w}_1; \ldots; \mathbf{w}_{n^2}] \in \mathbb{R}^{qn^2}$. We introduce (1.5) 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 (1.5) with respect to the third one has a closed-form formula that is easy to compute. In addition, this approach is numerical stable for large values of β and γ. Since $\mathbf{w}$ and z are decoupled for given u, our algorithm will minimize the objective function in (1.5) with respect to $(\mathbf{w}, z)$ and u, alternately. We show that, for any fixed $\beta, \gamma > 0$, this alternating minimization scheme generates a sequence of points converging to a solution of (1.5). 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.

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. 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 of 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 [23], the multistate median filter [16], and the median filter based on homogeneity information [18, 30]. 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 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 ω. To stabilize the recovery of $\bar{u}$, regularization is introduced, giving rise to the variational formulation

$$\min_u \Phi_{\text{reg}}(u) + \mu \Phi_{\text{fid}}(u, f), \tag{1.6}$$

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 [36], the TV regularization [31], 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 rotation of data in this case. In other cases, e.g., $\|\cdot\| = \|\cdot\|_1$, it is anisotropic. The isotropic TV is often preferred over any anisotropic ones. Compared with Tikhonov-like regularization, TV regularization has the advantage of preserving sharp edges and object boundaries. For multichannel images, TV has been extended in several ways such as "color-TV" in [5] and multichannel TV in [7, 11, 12, 35]. 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 of inverse problems; see e.g. [36, 31, 37]. 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 [27], while minimizing nonsmooth data fidelity will likely succeed. Theoretical results comparing smooth and nonsmooth data fidelity terms for image denoising can be found in [27, 28]. It is pointed out in [27] that nonsmooth fidelity terms can give correct pixel intensities under some favorable conditions.

The use of $\Phi_{\text{fid}}(u, f) = \|Ku - f\|_1$, which is nonsmooth, was introduced in [1]. Recently, geometric properties of the TVL1 model, which uses TV with the above L_1 fidelity term, are analyzed in [9, 41, 42]. Motivated by [9], the authors of [41, 42] 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 [33], independent of their locations or intensities. These results have led to the applications of the TVL1 model in medical imaging [40, 13] and computer vision [15, 14], 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-stage approach was proposed in [10, 8] where denoising (and possibly deblurring) is performed in two stages. In the first stage, the likely outliers are 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 uses 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. [37, 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 [38].

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 advantages of simple high-dimensional *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 (1.5) are studied. In Section 3, main convergence results of the proposed algorithm are established. Practical implementation of the proposed algorithm and a comparison to the method proposed in [20] are given in Section 4. In this section, multichannel image recovery results are also presented. Finally, conclusion remarks are given in Section 5.

2. Basic algorithm and optimality. Before deriving the algorithm, we introduce our notation. 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 unless otherwise specified. Additional notation will be introduced as the paper progresses.

2.1. An alternating minimization algorithm. To simplify analysis, we rewrite (1.4) and (1.5) by rescaling some quantities. Let

$$\tau = \sqrt{\mu\beta/\gamma} \quad \text{and} \quad (K, f, z) \leftarrow \sqrt{\mu\gamma/\beta}(K, f, z). \tag{2.1}$$

Then problems (1.4) and (1.5) are transformed to

$$\min_u \sum_i \alpha_i \|G_i u\| + \tau \|Ku - f\|_1, \tag{2.2}$$

and

$$\min_{\mathbf{w}, z, u} \sum_i \left(\alpha_i \|\mathbf{w}_i\| + \frac{\beta}{2}\|\mathbf{w}_i - G_i u\|^2 \right) + \tau\|z\|_1 + \frac{\beta}{2}\|z - (Ku - f)\|^2, \tag{2.3}$$

respectively. 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 (2.3). In addition, for all subscripts i the first two terms in (2.3) 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 (2.3). First, for a fixed u, the minimizer function of $\mathbf{w}_i$ is given by a multi-dimensional shrinkage [39]:

$$\mathbf{w}_i = \max\left\{ \|G_i u\| - \frac{\alpha_i}{\beta}, 0 \right\} \frac{G_i u}{\|G_i u\|}, \; i = 1, \ldots, n^2, \tag{2.4}$$

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:

$$z = \max\left\{ |Ku - f| - \frac{\tau}{\beta}, 0 \right\} \circ \operatorname{sgn}(Ku - f), \tag{2.5}$$

where "$\circ$" represents the pointwise product and all other operations are implemented componentwise. Clearly, the computational cost for (2.4) is linear in terms of qn^2, and that for (2.5) is linear in terms of n^2. Secondly, for fixed $\mathbf{w}$ and z, the minimization with respect to u is a least squares problem

$$\min_u \sum_i \|\mathbf{w}_i - G_i u\|^2 + \|Ku - (f + z)\|^2. \tag{2.6}$$

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

$$G \triangleq \begin{pmatrix} G^{(1)} \\ \vdots \\ G^{(q)} \end{pmatrix} \in \mathbb{R}^{qn^2 \times mn^2} \quad \text{and} \quad \mathbf{W} \triangleq \begin{pmatrix} \mathbf{w}_1^\top \\ \vdots \\ \mathbf{w}_{n^2}^\top \end{pmatrix} \triangleq [w_1; \ldots; w_q] \in \mathbb{R}^{n^2 \times q}, \tag{2.7}$$

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 $G_i = I_3 \otimes D_i$, then

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

In the above notation, the normal equations of (2.6) can be written as

$$\left[G^\top G + K^\top K\right] u = G^\top w + K^\top (f + z). \tag{2.8}$$

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. [26]. Therefore, both G and K can be blockwise diagonalized by pre- and post-multiplying 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 following two matrices,

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

are both blockwise diagonal matrices with the block size $n^2 \times n^2$. Then equation (2.8) is equivalent to

$$\left[\hat{G}^\top \hat{G} + \hat{K}^\top \hat{K}\right] (\mathbf{F}_m u) = \hat{G}^\top \mathbf{F}_q w + \hat{K}^\top (\mathbf{F}_m f + \mathbf{F}_m z), \tag{2.9}$$

where $\mathbf{F}_m v$ $(v = u, f, z)$ is the discrete Fourier transform of v, and similarly for $\mathbf{F}_q w$. Since both $\hat{G}$ and $\hat{K}$ are blockwise diagonal, the coefficient matrix in the left-hand side of (2.9) is also blockwise diagonal with $m \times m$ blocks of the size $n^2 \times n^2$.

The solution of (2.6) can be obtained by solving its normal equations (2.8) in three steps. First, we apply discrete FFTs to both sides of (2.8). Then, we solve the resulting blockwise diagonal systems (2.9) 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 u.

Clearly, all matrices involved in (2.8) need to be blockwise diagonalized only once. At every iteration, it is only necessary to apply m Fourier transforms to z, q Fourier transforms to w and m inverse Fourier transforms to $\mathbf{F}_m u$. Therefore, the total number of two-dimensional discrete Fourier transforms (including inverse Fourier transforms) of size $n^2 \times n^2$ is $2m + q$ per-iteration. Moreover, block Gaussian elimination applied to (2.9) without pivoting, where the coefficient matrix has $m \times m$ blocks of square diagonal matrices of size n^2, requites about $(2/3)m^3 n^2$ arithmetic operations.

Since minimizing the objective function in (2.3) with respect to each variable is computationally inexpensive, we propose solving (2.3) by the following alternating minimization scheme:

```
ALGORITHM 1. Input $f$, $K$, $\mu > 0$, $\beta, \gamma \gg 0$ 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 (2.4) and (2.5), respectively.
        2) Given $\mathbf{w}$ and $z$, compute $u$ by solving (2.8).
    End Do
```

The stopping criterion of Algorithm 1 is specified in next subsection based on the optimality conditions of (2.3). More details of Algorithm 1 are discussed in Section 4.

2.2. Optimality conditions. Now, we derive optimality conditions of (2.2) and (2.3) 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*

$$\partial f(x) = \begin{cases} \{A^\top Ax/\|Ax\|\}, & if\ Ax \neq 0; \\ \left\{A^\top h : \|h\| \leq 1, h \in \mathbb{R}^p\right\}, & otherwise. \end{cases} \tag{2.10}$$

Proof. A simple proof of this proposition can be found in [38]. □

For $t \in \mathbb{R}$, the signum and the signum set-valued functions are defined as

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

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

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

$$\partial g(x) = \left\{B^\top \lambda : \lambda \in \mathrm{SGN}(Bx)\right\}. \tag{2.11}$$

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

$$\|By\|_1 - \|Bx\|_1 - (B^\top \lambda)^\top (y - x) = \sum_i \{|(By)_i| - |(Bx)_i| - \lambda_i[(By)_i - (Bx)_i]\}. \tag{2.12}$$

Since $\lambda \in \mathrm{SGN}(Bx)$, 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 $\eta^\top \xi > \alpha > \eta^\top (B^\top \lambda), \forall \lambda \in \mathrm{SGN}(Bx)$. Let $I^+ = \{i : (Bx)_i > 0\}$, $I^- = \{i : (Bx)_i < 0\}$ and $I^0 = \{i : (Bx)_i = 0\}$. It follows that

$$\eta^\top \xi > \alpha \geq \sup\left\{\eta^\top (B^\top \lambda) : \lambda \in \mathrm{SGN}(Bx)\right\} = \sum_{i \in I^+} (B\eta)_i - \sum_{i \in I^-} (B\eta)_i + \sum_{i \in I^0} |(B\eta)_i|. \tag{2.13}$$

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

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

The above inequality contradicts (2.13). 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 (2.3) if and only if the subdifferential of the objective at $(\mathbf{w}, z, u)$ contains the origin. In light of propositions 2.1 and 2.2 with A and B being identity matrices of appropriate orders, the optimality conditions of (2.3) are

$$\text{(2.14)} \quad \begin{cases} \alpha_i\mathbf{w}_i/\|\mathbf{w}_i\| + \beta(\mathbf{w}_i - G_iu) = 0, & i \in I_1 \triangleq \{i : \mathbf{w}_i \neq \mathbf{0}\}, \\ \|G_iu\| \le \alpha_i/\beta, & i \in I_2 \triangleq \{i : \mathbf{w}_i = \mathbf{0}\}, \\ \tau\,\mathrm{sgn}(z_i) + \beta[z_i - (Ku - f)_i] = 0, & i \in I_3 \triangleq \{i : z_i \neq 0\}, \\ |(Ku - f)_i| \le \tau/\beta, & i \in I_4 \triangleq \{i : z_i = 0\}, \end{cases}$$

$$\text{(2.15)} \quad G^\top(Gu - w) + K^\top(Ku - f - z) = 0.$$

We note that (2.15) is just another expression of (2.8). Our stopping criterion for Algorithm 1 is based on (2.14) and (2.15). Let

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

where $\{I_j : j = 1, 2, 3, 4\}$ are defined as in (2.14). Algorithm 1 is terminated once

$$\text{(2.16)} \quad Res \triangleq \max\left\{\max_{i\in I_1}\{\|r_1(i)\|\}, \max_{i\in I_2}\{r_2(i)\}, \max_{i\in I_3}\{|r_3(i)|\}, \max_{i\in I_4}\{r_4(i)\}, \|r_5\|_\infty\right\} \le \epsilon$$

is met, where Res measures the total residual and $\epsilon > 0$ is a prescribed tolerance.

Let K_{I_3}, K_{I_4} be the submatrices of K with rows corresponding to those indices in I_3 and I_4 respectively. Similarly, v_I represents the subvector of v corresponding to I. From (2.14) and the relation between G and G_i described in (2.7), eliminating $\mathbf{w}$ and z from (2.15) gives

$$\text{(2.17)} \quad \sum_{i\in I_1} \alpha_i G_i^\top \frac{G_iu}{\|G_iu\|} + \sum_{i\in I_2} G_i^\top h_i + \tau \begin{pmatrix} K_{I_3} \\ K_{I_4} \end{pmatrix}^\top \begin{pmatrix} \mathrm{sgn}(K_{I_3}u - f_{I_3}) \\ v_{I_4} \end{pmatrix} = 0,$$

where $h_i \triangleq \beta G_iu$ satisfies $\|h_i\| \le \alpha_i$, and $v_{I_4} = \beta(K_{I_4}u - f_{I_4})/\tau$ satisfies $\|v_{I_4}\|_\infty \le 1$. Let u^* be any solution of (2.2). Define

$$I_1^* \triangleq \{i : G_iu^* \neq 0\}, \;\; I_2^* \triangleq \{i : G_iu^* = 0\}, \;\; I_3^* \triangleq \{i : (Ku^* - f)_i \neq 0\} \;\text{ and }\; I_4^* \triangleq \{i : (Ku^* - f)_i = 0\}.$$

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

$$\sum_{i \in I_1^*} \alpha_i G_i^\top \frac{G_i u^*}{\|G_i u^*\|} + \sum_{i \in I_2^*} G_i^\top h_i^* + \tau \begin{pmatrix} K_{I_3^*} \\ K_{I_4^*} \end{pmatrix}^\top \begin{pmatrix} \operatorname{sgn}(K_{I_3^*} u^* - f_{I_3^*}) \\ v_{I_4^*} \end{pmatrix} = 0. \tag{2.18}$$

Equation (2.17) differs from (2.18) only in the index sets involved. As β and γ increase, I_1 and I_3 will converge to I_1^* and I_3^*, respectively.

3. Convergence analysis. It is well-known that 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 [29] for example). In this section, we establish convergence and a q-linear convergence result of the proposed algorithm for fixed penalty parameters β and γ.

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

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

For $v \in \mathbb{R}^N$, let $s_\tau(v) \triangleq (s_\tau(v_1); \dots; s_\tau(v_N)) \in \mathbb{R}^N$, i.e. s_τ 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 $v_1, \dots, v_q \in \mathbb{R}^N$, $S(v_1; \dots; v_q) : \mathbb{R}^{qN} \to \mathbb{R}^{qN}$ is defined as

$$S(v_1; \dots; v_q) \triangleq (s(\mathbf{t}_1); \dots; s(\mathbf{t}_N)), \text{ where } \mathbf{t}_i = [(v_1)_i; \dots; (v_q)_i] \in \mathbb{R}^q,\ i = 1, \dots, N. \tag{3.1}$$

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_τ 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 [38]. □

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

Assumption 1. $\mathcal{N}(G) \cap \mathcal{N}(K) = \{0\}$.

The following matrices are used in our analysis:

$$H = \begin{pmatrix} G \\ K \end{pmatrix} \text{ and } M = G^\top G + K^\top K = H^\top H.$$

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

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

and

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

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

Using the above notation, the iteration formulae (2.4), (2.5) and (2.8) can be expressed as

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

and

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

Since the objective function in (2.3) is convex, bounded below, and coercive (i.e., its value goes to infinity as $\|(w, z, u)\| \to \infty$), problem (2.3) has at least one minimizer $(v^*; u^*) = (w^*; z^*; u^*)$, which should satisfy the fixed-point equations

$$v^* = (w^*; z^*) = \Big(S(G^{(1)}u^*; \ldots; G^{(q)}u^*); s_\tau(Ku^* - f)\Big) = \hat{S} \circ \hat{h}(v^*) \tag{3.4}$$

and the equations

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

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*

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

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 non-expansive 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 [38]. 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 (2.3).*

Proof. The proof is similar to that of Theorem 3.4 in [38] and thus is omitted. □

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

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

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:

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

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

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*

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

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

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

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

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

where the first equality comes from the iteration of $\mathbf{w}_i$ in (2.4) 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,

$$\|z^{k+1} - z^*\|^2 = \|s_\tau \circ h^{(q+1)}(v^k) - s_\tau \circ h^{(q+1)}(v^*)\|^2 \le \|h^{(q+1)}(v^k) - h^{(q+1)}(v^*)\|^2. \tag{3.9}$$

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

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

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

$$\left(z_i^{k+1} - z_i^* \right)^2 = \left(s_\tau \circ h_i^{(q+1)}(v^k) - s_\tau \circ h_i^{(q+1)}(v^*) \right)^2 \le \left| h_i^{(q+1)}(v^k) - h_i^{(q+1)}(v^*) \right|^2. \tag{3.11}$$

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

$$\begin{aligned}(3.12)\qquad \left(z_i^{k+1} - z_i^*\right)^2 &= \left(s_\tau \circ h_i^{(q+1)}(v^k)\right)^2 = \left(|h_i^{(q+1)}(v^k)| - \tau\right)^2 \\ &\leq \left\{\left|h_i^{(q+1)}(v^k) - h_i^{(q+1)}(v^*)\right| - \left(\tau - |h_i^{(q+1)}(v^*)|\right)\right\}^2 \\ &\leq \left|h_i^{(q+1)}(v^k) - h_i^{(q+1)}(v^*)\right|^2 - \left(\tau - |h_i^{(q+1)}(v^*)|\right)^2 \\ &\leq \left|h_i^{(q+1)}(v^k) - h_i^{(q+1)}(v^*)\right|^2 - \omega_2^2,\end{aligned}$$

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,

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

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 $E = \{1, \ldots, (q+m)n^2\} \setminus L$ 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. $\|v_E^{k+1} - v_E^*\| \leq \sqrt{\rho(P_{EE})}\|v_E^k - v_E^*\|$;
2. $\|u^{k+1} - u^*\|_M \leq \sqrt{\rho(P_{EE})}\|u^k - u^*\|_M$;

for all k sufficiently large.

Proof. From (3.2 - 3.5) and the non-expansiveness of S and s_τ, we get

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

and

$$\begin{aligned}(3.15)\qquad \|v^{k+1} - v^*\|^2 &= \|w^{k+1} - w^*\|^2 + \|z^{k+1} - z^*\|^2 \\ &= \|S(G^{(1)}u^k; \ldots; G^{(q)}u^k) - S(G^{(1)}u^*; \ldots; G^{(q)}u^*)\|^2 + \|s_\tau(Ku^k - f) - s_\tau(Ku^* - f)\|^2 \\ &\leq \|G(u^k - u^*)\|^2 + \|K(u^k - u^*)\|^2 = \|H(u^k - u^*)\|^2.\end{aligned}$$

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

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

Since we are only interested 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

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

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

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

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 $\rho(P_{EE}) \leq \rho(P) = 1$.

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 (250 × 250), with different blurs and noise. The original images are given in Figure 4.1. Two types of impulsive noises 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 to the algorithm in [20] for grayscale image deblurring. We next present color image results, and finally summarize the performance of our algorithm.

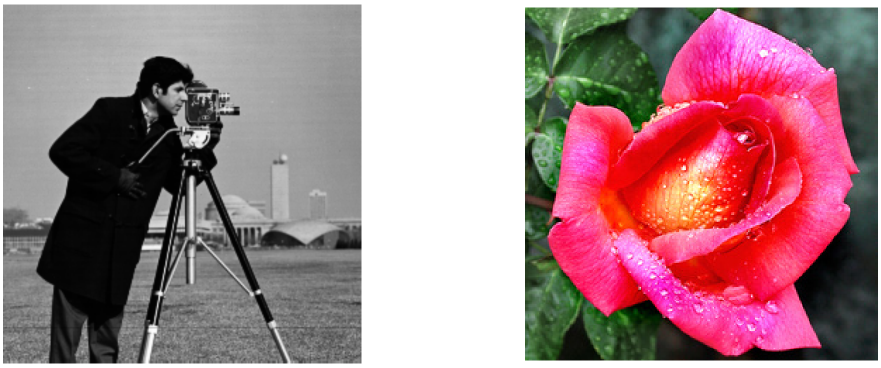

Fig. 4.1. *Original images. Cameraman (left, 256×256) and Rose (right, 250×250).*

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 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. As is usually done, we measured 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 regularizations based on TV, weight TV, and high-order derivatives (see [39] for details), we limited our experiments on model (1.3). The weighting parameter μ was determined experimentally. We limit $\beta = 2^{10}$ and $\gamma = 2^{15}$ in the approximation problem (1.5) which, based on our experimental results, are large enough to get almost the highest SNRs. Using $\beta > 2^{10}$ or $\gamma > 2^{15}$ would only increase computational cost but not solution quality. Similar settings were used in [38, 39].

After rescaling, μ and γ are hidden in K, f, z and τ (see (2.1) where $\tau = \sqrt{\mu\beta/\gamma}$). From (2.5), the smaller γ is, the larger τ is, and more zeros the shrinkage will produce. To speed up convergence, we implemented a continuation scheme on γ; that is, let γ take a small value at the beginning and gradually increase its value to 2^{15}. Specifically, we tested the γ-sequence $2^0, 2^1, 2^2 \dots, 2^{15}$. Accordingly, β was set to $2^0, 2^{2/3}, 2^{4/3} \dots, 2^{10}$. 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 β and γ 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 and γ from 1 to 2^{10} and 2^{15}, respectively. For fixed β and γ, the inner loop solves (1.5) until (2.16) is met. We set $\epsilon = 5 \times 10^{-3}$ as the default. 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 surprisingly well. Following [38], we give the name *fast total variation deconvolution*, or FTVd, to Algorithm 1 with the prescribed continuation scheme.

4.2. Grayscale image results. In this subsection, we compare the performance of FTVd with that of the algorithm proposed in [20], 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 [20] 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, 24, 34]. Specially, they used the factorized banded inverse preconditioner (FBIP, [25]), 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 support size of a blurring kernel 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 blurring kernels with a large support size, a balance needs to be found between the PCG iteration numbers and the cost of preconditioner computation; see [20] for details. Since the linear systems involved are increasingly ill-conditioned towards the end of outer iterations, a reasonable implementation of their algorithm should be using 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 [32] that is included in the LAD code provided to us by the authors of [20]) 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 only applicable 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 (2.4) to

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

where $|\cdot|$ represents componentwise absolute value. The optimality conditions (2.14)-(2.15) 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 next two paragraphs. Secondly, the LAD algorithm uses Neumann boundary conditions instead of 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 [26].

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 `std` = 5. For LAD, after we generated the blurry image by MATLAB function "`imfilter`" with symmetric boundary conditions, we corrupted 30% to 60% 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 with exactly the same salt-and-pepper noise recorded when generating noise for LAD. The blurry and noisy images and their restorations by FTVd and LAD are given in Figure 4.2, 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 only plotted the blurry and noisy images for FTVd.

As can be seen from Figure 4.2, the results of FTVd and LAD have similar qualities in all of the four tests. The results of LAD have slightly higher SNRs than those of FTVd but the differences are visually indistinguishable. By comparing the CPU times, 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 efforts 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 support size of blurring kernel became larger, the linear system that LAD needed to solve became even more ill-conditioned. As a result, the diagonal preconditioners used in [20] 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 support size of blurring kernels because it avoids solving any linear systems iteratively. To illustrate this, we applied the Gaussian blur of the size 15×15 and standard deviation `std` = 9 and the same levels of slat-and-pepper noise. The recovered results by FTVd are shown in Figure 4.3, from which we see that the CPU times required did not increase. However, we were not able to solve (4.1)

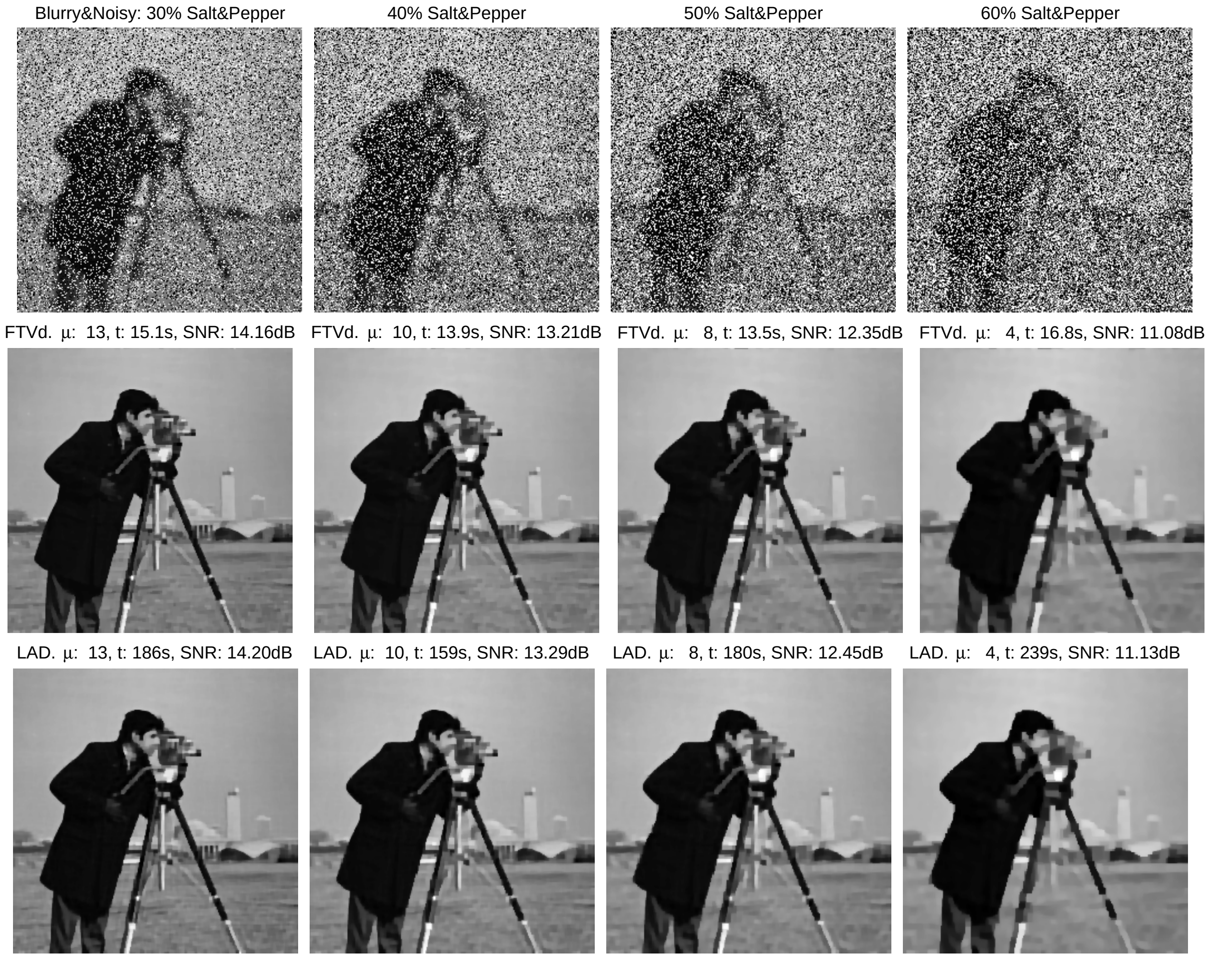
Blurry&Noisy: 30% Salt&Pepper
40% Salt&Pepper
50% Salt&Pepper
60% Salt&Pepper
FTVd. µ: 13, t: 15.1s, SNR: 14.16dB
FTVd. µ: 10, t: 13.9s, SNR: 13.21dB
FTVd. µ: 8, t: 13.5s, SNR: 12.35dB
FTVd. µ: 4, t: 16.8s, SNR: 11.08dB
LAD. µ: 13, t: 186s, SNR: 14.20dB
LAD. µ: 10, t: 159s, SNR: 13.29dB
LAD. µ: 8, t: 180s, SNR: 12.45dB
LAD. µ: 4, t: 239s, SNR: 11.13dB

Fig. 4.2. *Recovered from 7×7 sized Gaussian blur with salt-and-pepper noise from 30% to 60%.*

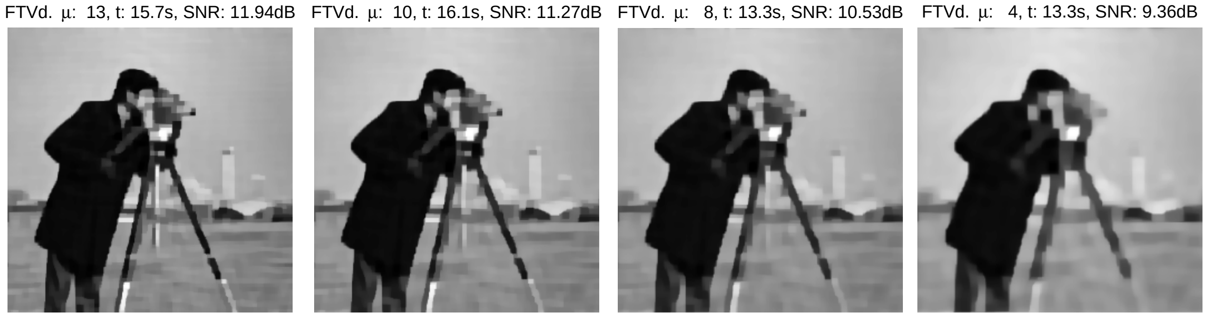
FTVd. µ: 13, t: 15.7s, SNR: 11.94dB
FTVd. µ: 10, t: 16.1s, SNR: 11.27dB
FTVd. µ: 8, t: 13.3s, SNR: 10.53dB
FTVd. µ: 4, t: 13.3s, SNR: 9.36dB

Fig. 4.3. *Recovered from 15×15 sized Gaussian blur with salt-and-pepper noise from 30% to 60% (left to right).*

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.

4.3. Color image results. In this subsection, we present recovery results for color images by FTVd. We first blurred the color image Rose by cross-channel blurring described below and then corrupted 30%

to 60% of its pixels at random with 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:

$$\begin{bmatrix} H_{rr} & H_{rg} & H_{rb} \\ H_{gr} & H_{gg} & H_{gb} \\ H_{br} & H_{bg} & H_{bb} \end{bmatrix} = \begin{bmatrix} 0.8\cdot(A,9) & 0.1\cdot(A,9) & 0.1\cdot(A,9) \\ 0.15\cdot(G,11,5) & 0.7\cdot(G,11,5) & 0.15\cdot(G,11,5) \\ 0.2\cdot(M,21,135) & 0.2\cdot(M,21,135) & 0.6\cdot(M,21,135) \end{bmatrix}, \tag{4.2}$$

where H_{σ_1,σ_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., [19, 21]. We note that the types, locations and support sizes of blurring kernels appear to have little or no influence on the efficiency of FTVd, as was also observed in [39]. 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.

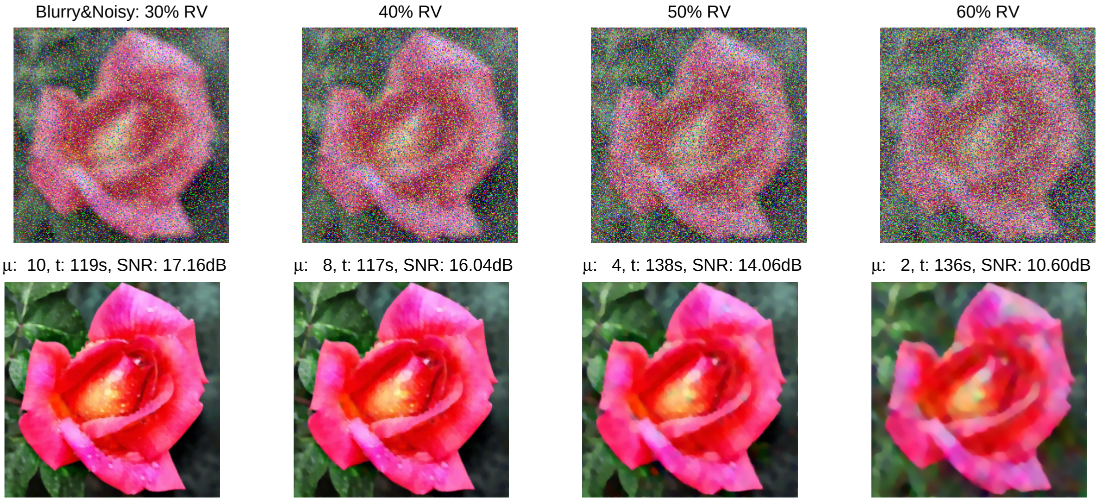
Blurry&Noisy: 30% RV
40% RV
50% RV
60% RV
µ: 10, t: 119s, SNR: 17.16dB
µ: 8, t: 117s, SNR: 16.04dB
µ: 4, t: 138s, SNR: 14.06dB
µ: 2, t: 136s, SNR: 10.60dB

Fig. 4.4. *Recovered from cross-channel blurring and random-valued noise from 30% to 60%.*

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 4.2 and 4.3 because the per-iteration computational cost has increased significantly from deblurring single-channel images to multichannel ones.

4.4. Summary. In the framework of Algorithm 1, there are two steps at each iteration. The first step computes $\mathbf{w}$ and z by (2.4) and (2.5), respectively, and the second solves (2.8) by FFTs and block Gaussian elimination. The computation of $\mathbf{w}$ and z only involves shrinkage operations and thus has linear complexity in terms of n^2. Therefore, the main computational work lies in solving the normal equations (2.8) by FFTs and block Gaussian elimination (for multichannel images). Consider $G_i = I_m \otimes D_i$ for all i. For grayscale images where $m = 1$ and $q = 2$, a total number of 4 FFTs (including 1 inverse FFT) are needed to solve

(2.8) while no Gaussian elimination is necessary. For color images where $m = 3$ and $q = 6$, a total number of 12 FFTs (including 3 inverse FFTs) are needed plus about $18n^2$ arithmetic operations required by the block Gaussian elimination. In our experiments, the total number of inner iterations required by FTVd was typically around 250. Therefore, the total number of FFTs was about 1000 for grayscale images and 3000 for RGB color images. Upon profiling our code, we observed that for color images about 40% of the total CPU time was spent on Gaussian elimination and checking optimality conditions; the rest was spent on FFTs and other calculations. Furthermore, the CPU time used by FTVd changes little with the size of the blurring kernel, and increases only moderately with the image size.

5. Conclusion 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 fast q-linear global convergence rate for the rest. At each iteration, the total computational cost is dominated by the costs of a number of FFTs and block Gaussian elimination. Our numerical results show that the algorithm is efficient and stable. For grayscale images and a relatively small kernel size, our algorithm is already over one order of magnitude faster than a state-of-the-art algorithm, and the advantage further widens as the kernel size increases. Our numerical results also show that while doing deblurring, the algorithm is capable of removing a large amount of impulsive noise from grayscale or color images that contaminates up to 60% pixels.

Acknowledgments. The authors would like to thank Dr. Haoying Fu for providing the code of [20]. The work of J. Yang has been supported by the Chinese Scholarship Council during his visit to Rice University. The work of Y. Zhang has been supported in part by NSF Grant DMS-0811188. The work of W. Yin has been supported in part by NSF CAREER Grant DMS-0748839.

REFERENCES

[1] S. Alliney, *Digital filters as absolute norm regularizers*, IEEE Trans. Signal Process., vol. 40, no. 6, pp. 1548–1562, 1992.

[2] J. Astola and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*, Boca Raton, FL: CRC, 1997.

[3] L. Bar, A. Brook, N. Sochen and N. Kiryati, *Deblurring of color images corrupted by impulsive noise*, IEEE Trans. Image Process., vol. 16, no. 4, pp. 1101–1110, 2007.

[4] M. Benzi, C. Meyer, and M. Tuma, *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput., 17, pp. 1135-1149, 1996.

[5] P. Blomgren and T. F. Chan, *Color TV: Total Variation Methods for Restoration of Vector-Valued Images*, IEEE Trans. Imag. Process., vol. 7, no. 3, pp. 304–309, 1998.

[6] A. Bovik, *Handbook of Image and Video Processing*, New York: Academic, 2000.

[7] A. Brook, R. Kimmel, and N. Sochen, *Variational restoration and edge detection for color images*, J. Math. Imag. Vis., vol. 18, pp. 247–268, 2003.

[8] J. F. Cai, R. Chan and M. Nikolova, *Two phase methods for deblurring images corrupted by impulse plus Gaussian noise*, AIMS Journal on Inverse Problems and Imaging, vol. 2, no. 2, pp. 187–204, 2008.

[9] T. F. Chan, and S. Esedoglu, *Aspects of total variation regularized L1 function approximation*, UCLA, Tech. Report, 2004.

[10] R. Chan, C. W. Ho and M. Nikolova, *Salt-and-Pepper Noise Removal by Median-type Noise Detector and Detail-preserving Regularization*, IEEE Trans. Imag. Process., vol. 14, no. 10, pp. 1479–1485, 2005.

[11] T. F. Chan, S. H. Kang, and J. Shen, *Total variation denoising and enhancement color images based on the CB and HSV color models*, Journal of Visual Communication and Image Representation, vol. 12, no. 4, 422–435, 2001.

[12] T. F. Chan, and J. Shen, *Variational restoration of non-flat image features: Models and algorithms*, SIAM J. Appl. Math. 61, 2000, 1338-1361.

[13] 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, vol. 3765 of Lecture Notes in Computer Science, Springer, 2005, pp. 114–124.

[14] T. Chen, W. Yin, X. S. Zhou, D. Comaniciu, and T. Huang, *Total variation models for variable lighting face recognition*, IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), vol. 28, pp. 1519–1524, 2006.

[15] 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, pp. 532–539, 2005.

[16] 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., vol. 48, no. 8, pp. 784789, 2001.

[17] R. Courant, *Variational methods for the solution of problems with equilibrium and vibration*, Bull. Amer. Math. Soc., 49, pp. 1–23, 1943.

[18] H. L. Eng and K. K. Ma, *Noise adaptive soft-switching median filter*, IEEE Trans. Image Process., vol. 10, no. 2, pp. 242-251, 2001.

[19] H. Y. Fu, M. K. Ng, and J. L. Barlow, *Structured total least squares for color image restoration*, SIAM J. Sci. Comput., vol. 28, no. 3, pp. 1100–1119, 2006.

[20] 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., vol. 27, no. 6, pp. 1881–1902, 2006.

[21] N. P. Galatsanos, A. K. Katsaggelos, R. T. Chan, and A. D. Hillery, *Least squares restorations of multichannel images*, IEEE Trans. Signal Process., vol. 39, pp. 2222–2236, 1991.

[22] D. Geman and C. Yang, *Nonlinear image recovery with half-quadratic regularization*, IEEE Trans. Image Process., vol. 4, pp. 932–946, 1995.

[23] H. Hwang and R. A. Haddad, *Adaptive median filters: New algorithms and results*, IEEE Trans. Image Process., vol. 4, no. 4, pp. 499502, 1995.

[24] L. Kolotilina, and A. Yeremin, *Factorized sparse approximate inverse preconditionings* I. *Theory*, SIAM J. Matrix Anal. Appl., vol. 14, pp. 45-58, 1993.

[25] F.-R. Lin, M. K. Ng, and W.-K. Ching, *Factorized banded inverse preconditioners for matrices with Toeplitz structure*, SIAM J. Sci. Comput., vol. 26, pp. 1852-1870, 2005.

[26] M. K. Ng, R. H.'Chan, and W. Tang, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., vol 21, no. 3, pp. 851–866, 1999.

[27] M. Nikolova, *Minimizers of cost-functions involving non-smooth data fidelity terms. Application to the processing of outliers*, SIAM J. Nume. Ana., vol. 40, no. 3, pp. 965–994, 2002.

[28] ———, *A variational approach to remove outliers and impulse noise*, Journal of Mathematical Imaging and Vision, vol. 20, no. 1-2, pp. 99–120, 2004.

[29] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 1999.

[30] G. Pok, J. C. Liu, and A. S. Nair, *Selective removal of impulse noise based on homogeneity level information*, IEEE Trans. Imag. Process., vol. 12, no. 1, pp. 85-92, 2003.

[31] L. I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, vol. 60, pp. 259–268, 1992.

[32] M. Saunders, *Interior methods for optimization with application to maximum entropy problems*, `http://sccm.stanford.edu/csri/Saunders.pdf`.

[33] O. Scherzer, W. Yin, and S. Osher, *Slope and G-set characterization of set-valued functions and applications to non-differentiable optimization problems*, Communications in Mathematical Sciences, vol. 3, pp. 479–492, 2005.

[34] W. Tang, *Toward an effective sparse approximate inverse preconditioner*, SIAM J. Matrix Anal. Appl., vol. 20, pp. 970-986, 1998.

[35] 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.

[36] A. Tikhonov, and V. Arsenin, *Solution of Ill-Posed problems*, Winston, Washington, DC, 1977.

[37] C. R. Vogel and M. E. Oman, *Fast, robust total variation based reconstruction of noisy, blurred images*, IEEE Trans. Image Process., vol. 7, pp. 813–824, 1998.

[38] Y. Wang, J. Yang, W. Yin and Y. Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, Accepted by SIAM J. Imag. Sci., 2008.

[39] J. Yang, W. Yin, Y. Zhang, and Y. Wang, *A fast algorithm for edge-preserving variational multichannel image restoration*, Tech. Report 08-09, CAAM, Rice University. Submitted.

[40] W. Yin, T. Chen, X. S. Zhou, and A. Chakraborty, *Background correction for cDNA microarray image using the $TV + L^1$ model*, Bioinformatics, vol. 21 pp. 2410–2416, 2005.

[41] 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, vol. 3752 of Leture Notes in Computer Science, Springer, pp. 73–84, 2005.

[42] ———, *The total variation regularized L^1 model for multiscale decomposition*, SIAM Journal on Multiscale Modeling and Simulation, vol. 6, pp. 190–211, 2006.