

A NEW ALTERNATING MINIMIZATION ALGORITHM FOR TOTAL VARIATION IMAGE RECONSTRUCTION

YILUN WANG*, JUNFENG YANG†, WOTAO YIN*, AND YIN ZHANG *

Abstract.

We propose, analyze and test an alternating minimization algorithm for recovering images from blurry and noisy observations with total variation (TV) regularization. This algorithm arises from a new half-quadratic model applicable to not only the anisotropic but also isotropic forms of total variation discretizations. The per-iteration computational complexity of the algorithm is three Fast Fourier Transforms (FFTs). We establish strong convergence properties for the algorithm including finite convergence for some variables and relatively fast exponential (or q -linear in optimization terminology) convergence for the others. Furthermore, we propose a continuation scheme to accelerate the practical convergence of the algorithm. Extensive numerical results show that our algorithm performs favorably in comparison to several state-of-the-art algorithms. In particular, it runs orders of magnitude faster than the Lagged Diffusivity algorithm for total-variation-based deblurring. Some extensions of our algorithm are also discussed.

Key words. Half-quadratic, image deblurring, isotropic total variation, Fast Fourier Transform

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

1. Introduction. In this paper, we propose a fast algorithm for reconstructing images from blurry and noisy observations. For simplicity, we assume that the underlying images have square domains, but all discussions can be equally applied to rectangle domains. Let $u^0 \in \mathbb{R}^{n^2}$ be an original $n \times n$ gray-scale image, $K \in \mathbb{R}^{n^2 \times n^2}$ represent a blurring (or convolution) operator, $\omega \in \mathbb{R}^{n^2}$ be additive noise, and $f \in \mathbb{R}^{n^2}$ an observation which satisfies the relationship:

$$(1.1) \quad f = Ku^0 + \omega.$$

Given f and K , the image u^0 is recovered from the model

$$(1.2) \quad \min_u \sum_{i=1}^{n^2} \|D_i u\| + \frac{\mu}{2} \|Ku - f\|_2^2,$$

where $D_i u \in \mathbb{R}^2$ denotes the discrete gradient of u at pixel i , and the sum $\sum \|D_i u\|$ is the discrete total variation (TV) of u (the issue of boundary conditions will be discussed later). The origin of (1.2) and related results will be briefly reviewed in Section 1.2. Model (1.2) is also widely referred to as the TV/ L^2 model.

In (1.2), the TV is *isotropic* if the norm $\|\cdot\|$ in the summation is 2-norm, and *anisotropic* if it is 1-norm. We emphasize here that unlike most previous work, our method applies to both isotropic and anisotropic TV. However, we will only treat the isotropic case, $\|\cdot\| = \|\cdot\|_2$, in detail because the treatment for the anisotropic case is completely analogous.

Our algorithm is derived from the well-known variable-splitting and penalty techniques in optimization; specifically, at each pixel an auxiliary variable $\mathbf{w}_i \in \mathbb{R}^2$ is introduced to transfer $D_i u$ out of the

*Department of Computational and Applied Mathematics, Rice University, 6100 Main, Houston, Texas, 77005, U.S.A. (yilun.wang, wotao.yin, yzhang@rice.edu)

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

non-differentiable term $\|\cdot\|_2$, and the difference between \mathbf{w}_i and $D_i u$ is penalized, yielding the following approximation model to (1.2):

$$(1.3) \quad \min_{\mathbf{w}, u} \sum_i \|\mathbf{w}_i\|_2 + \frac{\beta}{2} \sum_i \|\mathbf{w}_i - D_i u\|_2^2 + \frac{\mu}{2} \|Ku - f\|_2^2,$$

with a sufficiently large penalty parameter β . This type of quadratic penalty approach can be traced back to as early as Courant [13] in 1943. It is well known that the solution of (1.3) converges to that of (1.2) as $\beta \rightarrow \infty$. Clearly, the objective function in (1.3) is convex in (u, \mathbf{w}) . The motivation for this formulation is that while either one of the two variables u and \mathbf{w} is fixed, minimizing the function with respect to the other has a closed-form formula with low computational complexity and high numerical stability. We will show that, for any fixed β , the algorithm of minimizing u and \mathbf{w} alternately has finite convergence for some variables and q -linear convergence rates for the rest. Moreover, the overall convergence of this algorithm can be significantly accelerated by a theoretically well justified continuation approach on the penalty parameter β .

The objective function in (1.3) is quadratic in u and separable in \mathbf{w} ; therefore, problem (1.3) is *half-quadratic* according to Geman and Reynolds [17] and Geman and Yang [18]. Following the framework proposed in [17, 18], a number of half-quadratic models have been derived and analyzed. However, model (1.3) cannot be derived from (1.2) by the constructions given in [17, 18]; instead, a new approximate TV function must be first introduced before applying the construction of [18] (see Section 2.3 for more details). The approximate TV model (1.3) and the resulting alternating minimization algorithm were first proposed in [43] without a convergence analysis. A similar split method has recently been proposed that uses Bregman iterations [20].

We have implemented the proposed algorithm in MATLABTM and compared it with a C++ implementation of the Lagged Diffusivity algorithm [42], which is one of the fastest algorithms for solving (1.2). Numerical results presented in Section 5 show that our algorithm is much faster than the Lagged Diffusivity algorithm especially when the blurring kernel is relatively large. Compared with a few other deblurring algorithms which solve different models, including ForWaRD (a Fourier and wavelet shrinkage method [32]), MATLAB Image Processing Toolbox functions “`deconvwnr`” and “`deconvreg`”, our algorithm consistently generates higher quality images in comparable running times.

In the rest of this section, we will introduce the notation, give a brief review of regularization approaches, and then summarize the contributions and organization of this paper.

1.1. Notation. Let $D^{(j)} \in \mathbb{R}^{n^2 \times n^2}$, $j = 1, 2, 3, \dots$ be finite difference matrices and, in particular, $D^{(1)}$ and $D^{(2)}$ be the first order finite difference matrices in the horizontal and vertical directions, respectively. For $j \geq 3$, $D^{(j)}$ represents higher order finite difference matrices. For scalars α_i , vectors v_i and matrices A_i of appropriate sizes, $i = 1, 2$, we let $a = (\alpha_1; \alpha_2) \triangleq (\alpha_1, \alpha_2)^T$, $v = (v_1; v_2) \triangleq (v_1^T, v_2^T)^T$, and $A = (A_1; A_2) \triangleq (A_1^T, A_2^T)^T$. As is used in (1.2), $D_i \in \mathbb{R}^{2 \times n^2}$ is a two-row matrix formed by stacking the i -th rows of $D^{(1)}$ and $D^{(2)}$. Let $\rho(T)$ be the spectral radius of matrix T and $\mathcal{P}(\cdot)$ a projection operator. From here on, the norm $\|\cdot\|$ refers to the 2-norm unless otherwise indicated. Additional notation will be introduced as the paper progresses.

1.2. Existing Regularization Approaches. There are different approaches based on statistics [14, 5], Fourier and/or wavelet transforms [25, 31], or variational analysis [38, 7, 11] for image deblurring. Among them the simplest is the Maximum Likelihood estimation, which solves the least square problem $\min_u \|Ku - f\|^2$ and is also known as the inverse filter. However, the solution of this inverse filter, though best matches the probabilistic behavior of the data, is often unacceptable because the image deblurring problems are usually severely ill-conditioned and the observed data usually contains noise. To stabilize the restoration process, some *a priori* knowledge about the unknown image is utilized through the addition of a regularization term, resulting in the model

$$(1.4) \quad \min_u J(u) = \Phi_{reg}(u) + \frac{\mu}{2} \|Ku - f\|^2,$$

where $\Phi_{reg}(u)$ is the regularizer that enforces the priori knowledge and the parameter μ is used to balance the two terms.

Two classes of regularizers are well known. One is the Tikhonov-like class [41] including $\Phi_{reg}(u) = \sum_j \|D^{(j)}u\|^2$, where $D^{(j)}$'s stand for some finite difference operators. In these cases, since the resulting objective functions $J(u)$ are quadratic, it is relatively inexpensive to minimize them by solving linear systems of equations. However, since Tikhonov-like regularizers tend to make images overly smooth, they often fail to adequately preserve important image attributes such as sharp edges.

Another well-known class of regularizers are based on total variation (TV), which was first proposed for image denoising by Rudin, Osher and Fatemi in [39], and then extended to image deblurring in [38]. In comparison to Tikhonov-like regularizers, TV regularizers can better preserve sharp edges or object boundaries that are usually the most important features to recover. The TV/ L^2 model (1.2) has been widely studied (see, for example, [7, 8] and references thereof). The superiority of TV over Tikhonov-like regularization was analyzed in [1, 15] for recovering images containing piecewise smooth objects.

However, due to the non-differentiability and non-linearity of the TV functions, the TV/ L^2 model (1.2) is computationally more difficult to solve. Despite numerous efforts over the years, algorithms for solving the isotropic TV/ L^2 model were still much slower than those for solving Tikhonov-like regularization models. In this work, we aim to overcome this difficulty. The most important structure of the TV/ L^2 model is that both the blurring operator and the finite-difference operators can be treated as discrete convolutions under suitable boundary conditions. How to best exploit this structure is crucial for achieving high computational efficiency.

1.3. Contributions. This paper has three major contributions: the derivation of a new half-quadratic model and the accompanying algorithm, the analysis of the algorithm's convergence properties, and the evaluation of its practical performance.

1. To the best of our knowledge, (1.3) is the first half-quadratic method for the isotropic TV/ L^2 model whose leading computational requirement is three FFTs per iteration. Previously, this level of complexity was achievable only for the anisotropic case. Our extension not only enables better image quality, but more importantly can be generalized to color (or multi-channel) image reconstruction with TV regularization.
2. We established convergence results for our algorithm that are stronger than those proved for existing half-quadratic methods. We established finite convergence for some auxiliary variables and strong

q -linear convergence rates for the other variables. The q -linear rates are stronger than usual because they depend on the spectral radii of sub-matrices rather than that of the whole matrix as in usual cases.

3. Based on our convergence results, we propose a continuation scheme on the penalty parameter with warm start. Our extensive numerical results have confirmed that the proposed algorithm can be orders of magnitude faster than the lagged diffusivity method – one of the state-of-the-art methods for solving the TV/ L^2 model (1.2).

In addition, we can extend our method to other problems involving TV regularization such as compressive sensing. These extensions will be discussed in Section 6.

1.4. Organization. The rest of this paper is organized as follows. In section 2, we propose our alternating minimization algorithm, study optimality conditions, and describe its relations with previous works. Convergence properties of the proposed algorithm are analyzed in section 3. In section 4, we propose a continuation scheme, demonstrate its efficiency, and describe the implementation of our complete algorithm. Numerical results in comparison with the lagged diffusivity (LD) and some other typical image restoration methods are presented in section 5. In section 6, we discuss how our method can be extended to more general inverse problems including some compressed sensing formulations. Finally, concluding remarks are given in section 7.

2. Basic algorithm, optimality and related works. We first introduce two auxiliary vectors $w_1, w_2 \in \mathbb{R}^{n^2}$ to approximate $D^{(1)}u$ and $D^{(2)}u$, respectively, where we recall that $D^{(1)}, D^{(2)} \in \mathbb{R}^{n^2 \times n^2}$ represent the two first-order forward finite difference operators with appropriate boundary conditions. We denote $w = (w_1; w_2) \in \mathbb{R}^{2n^2}$ and $D = (D^{(1)}; D^{(2)}) \in \mathbb{R}^{2n^2 \times n^2}$, where the latter is the total finite difference operator. For $i = 1, \dots, n^2$, we let $\mathbf{w}_i = ((w_1)_i; (w_2)_i) \in \mathbb{R}^2$, which is an approximation of $D_i u = [(D^{(1)}u)_i; (D^{(2)}u)_i] \in \mathbb{R}^2$. To keep \mathbf{w}_i close to $D_i u$, we apply a quadratic penalty term, which is the second term in (1.3).

2.1. An alternating minimization algorithm. It is easy to minimize the objective function in (1.3) with respect to either u or \mathbf{w} . For a fixed u , the first two terms in (1.3) are separable with respect to \mathbf{w}_i , so minimizing (1.3) for \mathbf{w} is equivalent to solving, for $i = 1, 2, \dots, n^2$,

$$(2.1) \quad \min_{\mathbf{w}_i} \left(\|\mathbf{w}_i\| + \frac{\beta}{2} \|\mathbf{w}_i - D_i u\|^2 \right),$$

for which the unique minimizer is given by the following two-dimensional shrinkage formula:

$$(2.2) \quad \mathbf{w}_i = \max \left\{ \|D_i u\| - \frac{1}{\beta}, 0 \right\} \frac{D_i u}{\|D_i u\|}, \quad i = 1, \dots, n^2,$$

where the convention $0 \cdot (0/0) = 0$ is followed. For the anisotropic case $\|\cdot\| = \|\cdot\|_1$, each component of \mathbf{w}_i is given by the simpler 1D shrinkage formula: $\mathbf{w}_i = \max\{|D_i u| - 1/\beta, 0\} \text{sgn}(D_i u)$, where all operations are done component-wise. On the other hand, for a fixed \mathbf{w} , (1.3) is quadratic in u and the minimizer u is given by the normal equations

$$\left(\sum_i D_i^T D_i + \frac{\mu}{\beta} K^T K \right) u = \sum_i D_i^T \mathbf{w}_i + \frac{\mu}{\beta} K^T f,$$

or equivalently,

$$(2.3) \quad \left((D^{(1)})^T D^{(1)} + (D^{(2)})^T D^{(2)} + \frac{\mu}{\beta} K^T K \right) u = (D^{(1)})^T w_1 + (D^{(2)})^T w_2 + \frac{\mu}{\beta} K^T f.$$

Under the periodic boundary condition for u , $(D^{(1)})^T D^{(1)}$, $(D^{(2)})^T D^{(2)}$ and $K^T K$ are all block circulant (see [33, 21], for example). Therefore, the Hessian matrix on the left-hand side of (2.3) can be diagonalized by two-dimensional discrete Fourier transform \mathcal{F} . Using the convolution theorem of Fourier transforms, we can write

$$(2.4) \quad u = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(D^{(1)})^* \circ \mathcal{F}(w_1) + \mathcal{F}(D^{(2)})^* \circ \mathcal{F}(w_2) + (\mu/\beta) \mathcal{F}(K)^* \circ \mathcal{F}(f)}{\mathcal{F}(D^{(1)})^* \circ \mathcal{F}(D^{(1)}) + \mathcal{F}(D^{(2)})^* \circ \mathcal{F}(D^{(2)}) + (\mu/\beta) \mathcal{F}(K)^* \circ \mathcal{F}(K)} \right),$$

where “*” denotes complex conjugacy, “o” denotes component-wise multiplication, and the division is component-wise as well. With a slight abuse of notation, we have used $\mathcal{F}(K)$ for the Fourier transform of the function represented by K in the convolution Ku (and similarly for $D^{(1)}$ and $D^{(2)}$). Since all quantities but w_1 and w_2 are constant, computing u from (2.4) involves two FFTs and one inverse FFT, once the constant quantities are computed.

Alternatively, under the Neumann boundary condition and assuming that the blurring kernel K is symmetric, the left-hand side of (2.3) is a block Toeplitz-plus-Hankel matrix (see [33]), so (2.3) can be diagonalized by two-dimensional discrete cosine transforms (DCTs), and solving (2.3) requires three DCT calls. In our numerical experiments, we used the periodic boundary condition and FFTs.

Since minimizing the objective function in (1.3) with respect to either \mathbf{w} or u is computationally inexpensive, we solve (1.3) for a fixed β by an alternating minimization scheme given below.

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

While “not converged”, **Do**

- 1) *Compute* w according to (2.2) for fixed u .
- 2) *Compute* u according to (2.4) for fixed w .

End Do

More details about this algorithm will be presented later. Next, we derive the optimality conditions for (1.3), based on which we will specify the stopping criterion (2.10) for Algorithm 1.

2.2. Optimality conditions. To study the optimality conditions of (1.3) for a fixed β , we need the following proposition.

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

$$(2.5) \quad \partial f(x) = \begin{cases} \{A^T Ax / \|Ax\|\}, & \text{if } Ax \neq 0; \\ \{A^T h : \|h\| \leq 1, h \in \mathbb{R}^p\}, & \text{otherwise.} \end{cases}$$

Proof. By the definition of sub-differential for a convex function,

$$(2.6) \quad \partial f(x) \triangleq \{g \in \mathbb{R}^n : \|Ay\| - \|Ax\| \geq g^T(y - x), \forall y \in \mathbb{R}^n\}.$$

If $\|Ax\| \neq 0$, then f is differentiable at x and clearly (2.5) holds. Now assume $\|Ax\| = 0$. In this case, after setting $y = x + d$, we have

$$\partial f(x) = \{g \in \mathbb{R}^n : \|Ad\| \geq g^T d, \forall d \in \mathbb{R}^n\}.$$

We will show that $\partial f(x) \equiv \mathcal{S} \triangleq \{A^T h : \|h\| \leq 1, h \in \mathbb{R}^p\}$. First, for any $A^T h \in \mathcal{S}$, $(A^T h)^T d \leq \|Ad\|$ by Cauchy-Schwarz inequality, implying $\mathcal{S} \subset \partial f(x)$. Next, we show $\partial f(x) \subset \mathcal{S}$ by contradiction. Suppose that $g \in \partial f(x)$ but $g \notin \mathcal{S}$. Since \mathcal{S} is closed and convex, by the well-known separation theorem, there exists $d \in \mathbb{R}^n$ and $\tau \in \mathbb{R}$ such that the hyper-plane $d^T x = \tau$ separates g and \mathcal{S} so that $d^T g > \tau > d^T p, \forall p \in \mathcal{S}$. It follows that

$$d^T g > \tau \geq \sup\{d^T A^T h : \|h\| \leq 1, h \in \mathbb{R}^p\} = \|Ad\|,$$

contradicting the assumption that $g \in \partial f(x)$. Therefore, $\partial f(x) = \mathcal{S}$ and the result is proved. \square

Since the objective function in (1.3) is convex, a pair (\mathbf{w}, u) solves (1.3) if and only if the sub-differential of the objective function at (\mathbf{w}, u) contains the origin, which gives the the following optimality conditions in light of Proposition 2.1 with $A = I$,

$$(2.7) \quad \begin{cases} \mathbf{w}_i / \|\mathbf{w}_i\| + \beta(\mathbf{w}_i - D_i u) = 0, & i \in I_1 \triangleq \{i : \mathbf{w}_i \neq \mathbf{0}\}, \\ \beta \|D_i u\| \leq 1, & i \in I_2 \triangleq \{i : \mathbf{w}_i = \mathbf{0}\}, \end{cases}$$

$$(2.8) \quad \beta D^T (Du - w) + \mu K^T (Ku - f) = 0.$$

Our stopping criterion for Algorithm 1 is based on the optimality conditions (2.7) and (2.8). Let

$$(2.9) \quad \begin{cases} r_1(i) \triangleq (\mathbf{w}_i / \|\mathbf{w}_i\|) / \beta + \mathbf{w}_i - D_i u, & i \in I_1, \\ r_2(i) \triangleq \|D_i u\| - 1 / \beta, & i \in I_2, \\ r_3 \triangleq \beta D^T (Du - w) + \mu K^T (Ku - f), \end{cases}$$

where I_1 and I_2 are defined as in (2.7). Algorithm 1 is terminated once

$$(2.10) \quad Res \triangleq \max \left\{ \max_{i \in I_1} \{\|r_1(i)\|\}, \max_{i \in I_2} \{r_2(i)\}, \|r_3\|_\infty \right\} \leq \epsilon,$$

where Res measures the total residual and $\epsilon > 0$ is a prescribed tolerance. In (2.9), condition (2.7) is scaled by a factor of $1/\beta$, but (2.8) is not, because in practice the latter can be met quickly even without this scaling.

Combining (2.7) and (2.8) to eliminate the w -variable, we can derive

$$(2.11) \quad \sum_{i \in I_1} D_i^T \frac{D_i u}{\|D_i u\|} + \sum_{i \in I_2} D_i^T h_i + \mu K^T (Ku - f) = 0,$$

where $h_i \triangleq \beta D_i u$ that satisfies $\|h_i\| \leq 1$. Let u^* be any solution of (1.2). Define $I_1^* = \{i, D_i u^* \neq 0\}$ and $I_2^* = \{1, \dots, n^2\} \setminus I_1^*$. Then, from proposition 2.1, there exist $\{h_i^* \in \mathbb{R}^2 : \|h_i^*\| \leq 1, i \in I_2^*\}$ such that

$$(2.12) \quad \sum_{i \in I_1^*} D_i^T \frac{D_i u^*}{\|D_i u^*\|} + \sum_{i \in I_2^*} D_i^T h_i^* + \mu K^T (Ku^* - f) = 0.$$

Equation (2.11) differs from (2.12) only in the index sets over which summations are taken. As β increases, I_1 should approach I_1^* . In Section 3, we will study the convergence properties of Algorithm 1.

2.3. Related works. While (1.3) is an instance of the classical quadratic penalty method, it is also a half-quadratic model according to Geman and Reynolds [17] and Geman and Yang [18]. However, this model

cannot be directly derived from the original frameworks in [17, 18] without introducing a new approximation to the isotropic TV function.

Consider the following general framework of recovering an image u from its corrupted measurements f :

$$(2.13) \quad \min_u \sum_i \phi(g_i^T u) + \frac{\mu}{2} \|Ku - f\|^2,$$

where $g_i \in \mathbb{R}^{n^2}$ is a local finite difference operator, $\phi(g_i^T \cdot)$ is convex and edge-preserving, and K is a convolution operator. Instead of solving (2.13) directly, the authors of [17] and [18] propose solving the equivalent problem

$$(2.14) \quad \min_{u,b} \sum_i (Q(g_i^T u, b_i) + \psi(b_i)) + \frac{\mu}{2} \|Ku - f\|^2$$

where $Q(t, s)$ and $\psi(s)$ are chosen such that $Q(t, s)$ is quadratic in t and

$$(2.15) \quad \phi(t) = \min_{s \in \mathbb{R}} (Q(t, s) + \psi(s)), \quad \forall t \in \mathbb{R}.$$

The name ‘‘half-quadratic’’ comes from the fact that the objective function in (2.14) is quadratic in u , but not in b . As such, under certain conditions such as convexity (2.14) can be solved by minimizing with respect to u and b alternately. Computationally, it is important that the constructed Q and ψ allow fast solutions for u and b , respectively. For this purpose, two forms of half-quadratic formulations have been widely studied: *multiplicative* form [17] in which $Q(t, s) = \frac{1}{2}t^2s$ and *additive* form [18] in which $Q(t, s) = \frac{1}{2}(t - s)^2$. For comparisons of these two forms of methods both in theory and in practice, see [24, 2, 34, 12] and references therein.

Unfortunately, the constructions given in [17] and [18] cannot be directly applied to either the anisotropic or the isotropic TV because some technical conditions fail to hold. Over the years, numerous half-quadratic models have been proposed to approximate the TV functions. Some of these models are of the additive form [24, 34], while others of the multiplicative form [7, 24]. However, in the half-quadratic models of the multiplicative form, the partial Hessians with respect to u depend on b , and thus vary from one iteration to another. In addition, such partial Hessians do not have a block circulant structure. As a result, half-quadratics of the multiplicative form cannot be efficiently minimized by FFTs.

Our half-quadratic model in (1.3) approximates the isotropic TV function, and is of the additive form. The Hessian of the quadratic is constant and has a block circulant structure (under appropriate boundary conditions). The only other half-quadratic model for TV approximation that shares these attributes is the one by Nikolova and Ng [34], but it is for the *anisotropic* rather than *isotropic* TV function. In [34], the authors use the Huber function

$$(2.16) \quad \phi(t) = \begin{cases} \frac{1}{2\epsilon}t^2, & \text{if } |t| \leq \epsilon; \\ |t| - \frac{\epsilon}{2}, & \text{otherwise,} \end{cases}$$

where $0 < \epsilon \ll 1$, to approximate $\phi(t) = |t|$ in (2.13) and apply the additive form half-quadratic construction. The resulting problem (2.14) is quadratic in u and separable in b , and the Hessian of the quadratic is constant and has a block circulant structure, enabling the use of fast transforms. However, [34] does not include a convergence analysis or extensive numerical experiments on the half-quadratic model for this anisotropic

TV approximation. In [34], two approaches are proposed to approximate the isotropic TV function. One is based on the multiplicative construction; the other employs the additive construction but does not yield a half-quadratic model. Neither approach leads to a fast alternating minimization algorithm.

Thus far all of the edge preserving functions ϕ used in half-quadratic constructions that we are aware of take only scalar variables. In order to approximate the isotropic TV function, one should allow ϕ to take vector variables; then our model (1.3) can be derived from a similar additive construction, which we describe below. First, (1.2) is approximated by

$$(2.17) \quad \min_u \sum_i \phi(D_i u) + \frac{\mu}{2} \|Ku - f\|^2,$$

where $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a 2D Huber-like function defined as

$$\phi(\mathbf{t}) = \begin{cases} \frac{\beta}{2} \|\mathbf{t}\|^2, & \text{if } \|\mathbf{t}\| \leq 1/\beta; \\ \|\mathbf{t}\| - \frac{1}{2\beta}, & \text{otherwise,} \end{cases}$$

for $\beta \gg 0$. Then a similar additive construction in 2D, instead of in 1D, gives $Q(\mathbf{t}, \mathbf{s}) = \frac{\beta}{2} \|\mathbf{t} - \mathbf{s}\|^2$ and $\psi(\mathbf{s}) = \|\mathbf{s}\|$ where $\mathbf{s}, \mathbf{t} \in \mathbb{R}^2$, which satisfy the 2D version of (2.15). Clearly, $\|\mathbf{s}\|$ and $\frac{\beta}{2} \|\mathbf{t} - \mathbf{s}\|^2$ are nothing but the first and second terms in our model (1.3) for $\mathbf{s} = \mathbf{w}_i$ and $\mathbf{t} = D_i u$. Therefore, (1.3) turns out to be a long overdue extension to Nikolova and Ng's half-quadratic model [34].

Since the work of Geman and Reynolds [17] and Geman and Yang [18], the convergence of alternating minimization algorithms for various half-quadratic models has been extensively studied, including that for multiplicative formulations in [24, 2, 34, 12] and for additive formulations in [24, 2, 34]. However, these analyses were done under some strict convexity assumptions on objective functions, see [24, 34, 2, 12]. Moreover, the convergence rate results obtained were r -linear at best; see [34, 2] for example. Unlike these previous results, our convergence results, presented in the next section, require neither nonsingularity of the operator K , nor strict convexity of the objective function as a whole (notice that $\phi = \|\cdot\|_2$ is not strictly convex). Moreover, we establish a finite convergence property for some variables and strong q -linear convergence for the rest.

3. Convergence Analysis. The convergence of the quadratic penalty method as the penalty parameter goes to infinity is well known (see Theorem 17.1 in [35] for example). That is, as $\beta \rightarrow \infty$, the solution of (1.3) converges to that of (1.2). However, it is generally difficult to determine theoretically how large a β value is enough for attaining a given accuracy. We will later choose β based on numerical experiments.

In this section, we analyze the convergence properties of Algorithm 1 for a fixed $\beta > 0$. First, we prove that the sequence $\{(w^k, u^k)\}$ generated by Algorithm 1 from any initial point converges to a solution of (1.3). Then, using a finite convergence property for $\{w^k\}$ we establish q -linear convergence rates for Algorithm 1. In what follows, we omit β for notational simplicity.

For $a \in \mathbb{R}^2$, the 2D shrinkage operator $s : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is defined as

$$(3.1) \quad s(a) \triangleq \max \left\{ \|a\| - \frac{1}{\beta}, 0 \right\} \frac{a}{\|a\|},$$

where the convention $0 \cdot (0/0) = 0$ is followed. It is easy to see that

$$s(a) = a - \mathcal{P}(a),$$

where $\mathcal{P}(\cdot) \triangleq \mathcal{P}_{\mathcal{B}}(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is the projection onto the closed disc $\mathcal{B} \triangleq \{a \in \mathbb{R}^2 \mid \|a\| \leq 1/\beta\}$. For vectors $u, v \in \mathbb{R}^N$, $N \geq 1$, we define $S(u; v) : \mathbb{R}^{2N} \rightarrow \mathbb{R}^{2N}$ by

$$S(u; v) \triangleq (s(a_1); \cdots; s(a_N)), \quad \text{where } a_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix},$$

i.e., S applies 2D shrinkage to each pair $(u_i, v_i) \in \mathbb{R}^2$, for $i = 1, 2, \dots, N$.

The first convergence result is Theorem 3.4, and we take a few steps to prove it. The first step is to prove the non-expansiveness of the shrinkage operator s .

The following proposition is known to hold for projections \mathcal{P} onto any closed convex set. We include a proof for the sake of completeness.

PROPOSITION 3.1. *For any $a, b \in \mathbb{R}^2$, it holds that*

$$\|s(a) - s(b)\|^2 \leq \|a - b\|^2 - \|\mathcal{P}(a) - \mathcal{P}(b)\|^2.$$

Furthermore, if $\|s(a) - s(b)\| = \|a - b\|$, then $s(a) - s(b) = a - b$.

Proof. Since \mathcal{B} is convex, $\mathcal{P}(\cdot)$ satisfies

$$(a - \mathcal{P}(a))^T (\mathcal{P}(a) - \mathcal{P}(b)) \geq 0, \quad \forall a, b \in \mathbb{R}^2.$$

Exchanging a and b gives us $-(b - \mathcal{P}(b))^T (\mathcal{P}(a) - \mathcal{P}(b)) \geq 0$. Adding the two yields

$$(3.2) \quad (a - b)^T (\mathcal{P}(a) - \mathcal{P}(b)) \geq \|\mathcal{P}(a) - \mathcal{P}(b)\|^2.$$

Therefore, from $s(a) = a - \mathcal{P}(a)$, we get

$$\begin{aligned} \|s(a) - s(b)\|^2 &= \|a - b - (\mathcal{P}(a) - \mathcal{P}(b))\|^2 \\ &= \|a - b\|^2 - 2(a - b)^T (\mathcal{P}(a) - \mathcal{P}(b)) + \|\mathcal{P}(a) - \mathcal{P}(b)\|^2 \\ &\leq \|a - b\|^2 - \|\mathcal{P}(a) - \mathcal{P}(b)\|^2. \end{aligned}$$

where the last inequality follows from (3.2). Moreover, if $\|s(a) - s(b)\| = \|a - b\|$, then $\mathcal{P}(a) = \mathcal{P}(b)$ and $s(a) - s(b) = a - b - (\mathcal{P}(a) - \mathcal{P}(b)) = a - b$. \square

We will make use of the following assumption in our convergence analysis. It is a mild assumption that has been used in most, if not all, previous analysis of the similar type.

ASSUMPTION 1. $\mathcal{N}(K) \cap \mathcal{N}(D) = \{0\}$, where $\mathcal{N}(\cdot)$ represents the null space of a matrix.

The following two symmetric positive definite matrices will be used in our analysis:

$$(3.3) \quad M = D^T D + \frac{\mu}{\beta} K^T K \quad \text{and} \quad T = DM^{-1}D^T.$$

Assumption 1 ensures the non-singularity of M , thus T is well defined. It is worth noting that $\rho(T) \leq 1$. We also define a linear operator $h : \mathbb{R}^{2n^2} \rightarrow \mathbb{R}^{2n^2}$ as $h(w) = (h^{(1)}(w); h^{(2)}(w))$, where

$$h^{(j)}(w) = D^{(j)} M^{-1} \left(D^T w + \frac{\mu}{\beta} K^T f \right), \quad j = 1, 2.$$

Using the definitions of S and h , we can rewrite the two iterative steps in Algorithm 1 into:

$$(3.4) \quad w^{k+1} = S(D^{(1)} u_k; D^{(2)} u_k) = S \circ h(w^k),$$

$$(3.5) \quad u^{k+1} = M^{-1} \left(D^T w^{k+1} + \frac{\mu}{\beta} K^T f \right).$$

Since the objective function in (1.3) is convex, bounded below, and coercive (i.e., its value goes to infinity as $\|(w, u)\| \rightarrow \infty$), (1.3) has at least one minimizer pair (w^*, u^*) that cannot be decreased by our alternating minimization scheme and thus must satisfy

$$(3.6) \quad w^* = S(D^{(1)}u^*; D^{(2)}u^*) = S \circ h(w^*),$$

$$(3.7) \quad u^* = M^{-1} \left(D^T w^* + \frac{\mu}{\beta} K^T f \right).$$

Particularly, (3.6) means that w^* is a fixed-point of $S \circ h$. We need to show that h is non-expansive.

PROPOSITION 3.2. *For any $w \neq \tilde{w}$ in \mathbb{R}^{2n^2} , it holds that*

$$\|h(w) - h(\tilde{w})\| \leq \|w - \tilde{w}\|$$

and the equality holds if and only if $h(w) - h(\tilde{w}) = w - \tilde{w}$.

Proof. From the definitions of M and T , it can be shown that $\rho(T) = \rho(DM^{-1}D^T) \leq 1$ and

$$\|h(w) - h(\tilde{w})\| = \|T(w - \tilde{w})\| \leq \rho(T)\|w - \tilde{w}\| \leq \|w - \tilde{w}\|.$$

Let $T = Q^T \Lambda Q$ be the eigen-decomposition of T , where Q is an orthogonal matrix and Λ is a diagonal matrix with elements $0 \leq \lambda_i \leq 1, \forall i$. If $\|h(w) - h(\tilde{w})\| = \|w - \tilde{w}\|$, namely, $\|Q^T \Lambda Q(w - \tilde{w})\| = \|w - \tilde{w}\|$, then $\|\Lambda Q(w - \tilde{w})\| = \|Q(w - \tilde{w})\|$. Since Λ is diagonal and $0 \leq \lambda_i \leq 1$, $\Lambda Q(w - \tilde{w}) = Q(w - \tilde{w})$ holds true. Multiplying both sides by Q^T completes the proof. \square

The following lemma gives a useful property for fixed points of the operator $S \circ h$.

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

Proof. From propositions 3.1, 3.2 and the definition of S , it holds $\|S \circ h(w) - S \circ h(\hat{w})\| \leq \|w - \hat{w}\|$. If equality holds, then again from from propositions 3.1, 3.2, we get

$$S \circ h(w) - S \circ h(\hat{w}) = h(w) - h(\hat{w}) = w - \hat{w}.$$

Since $\hat{w} = S \circ h(\hat{w})$, we get $w = S \circ h(w)$. \square

Now we are ready to prove the convergence of Algorithm 1.

THEOREM 3.4. *For any fixed $\beta > 0$, the sequence $\{(w^k, u^k)\}$ generated by Algorithm 1 from any starting point (w^0, u^0) converges to a solution (w^*, u^*) of (1.3).*

Proof. We prove the convergence of $\{w^k\}$ in three steps. First, from (3.4), the non-expansiveness of S and h , it is easy to show that the sequence $\{w^k\}$ lies in a compact region and thus has at least one limit point, say $w^* = \lim_{j \rightarrow \infty} w^{k_j}$. Let \hat{w} be any fixed point of $S \circ h$, i.e. $\hat{w} = S \circ h(\hat{w})$ and we get

$$\|w^k - \hat{w}\| = \|S \circ h(w^{k-1}) - S \circ h(\hat{w})\| \leq \|w^{k-1} - \hat{w}\|.$$

Therefore, the following limit exists,

$$(3.8) \quad \lim_{k \rightarrow \infty} \|w^k - \hat{w}\| = \lim_{j \rightarrow \infty} \|w^{k_j} - \hat{w}\| = \|w^* - \hat{w}\|,$$

which means that all limit points of $\{w^k\}$, if more than one, have an equal distance to \hat{w} . By the continuity of $S \circ h$, we have

$$S \circ h(w^*) = \lim_{j \rightarrow \infty} S \circ h(w^{k_j}) = \lim_{j \rightarrow \infty} w^{k_j+1}.$$

Hence, $S \circ h(w^*)$ is also a limit point of $\{w^k\}$ that must have an equal distance to \hat{w} as w^* does; i.e.,

$$\|w^* - \hat{w}\| = \|S \circ h(w^*) - \hat{w}\| = \|S \circ h(w^*) - S \circ h(\hat{w})\|.$$

It follows from lemma 3.3 that $w^* = S \circ h(w^*)$. Since \hat{w} is any fixed point of $S \circ h$, by replacing \hat{w} with w^* in (3.8), we establish the convergence of $\{w^k\}$: $\lim_{k \rightarrow \infty} w^k = w^*$. The convergence of $\{u^k\}$ to some u^* follows directly from (3.5). Therefore, (w^*, u^*) satisfies (3.6)-(3.7) and is a solution of (1.3). \square

We note that Theorem 3.4 does not require the objective function in (1.3) to be strictly convex in (u, w) (even though it is so with respect to each individual variable). As such, (1.3) may have multiple solution, and the limit (w^*, u^*) of the sequence generated by Algorithm 1 depends on starting point (w^0, u^0) .

Next we develop a finite convergence property for the auxiliary variable w . Let

$$h_i(w) = (h_i^{(1)}(w); h_i^{(2)}(w)) \in \mathbb{R}^2, \quad i = 1, \dots, n^2;$$

namely, $h_i(w)$ is a vector formed by stacking the i -th components of $h^{(1)}(w)$ and $h^{(2)}(w)$. We will make use of the following two index sets:

$$(3.9) \quad L = \left\{ i : \|D_i u^*\| = \|h_i(w^*)\| < \frac{1}{\beta} \right\} \quad \text{and} \quad E = \{1, \dots, n^2\} \setminus L.$$

THEOREM 3.5 (Finite convergence). *The sequence $\{(w^k, u^k)\}$ generated by Algorithm 1 from any starting point (w^0, u^0) satisfies $\mathbf{w}_i^k = \mathbf{w}_i^* = 0, \forall i \in L$, for all but a finite number of iterations that does not exceed $\|w^0 - w^*\|^2 / \omega^2$, where*

$$(3.10) \quad \omega \triangleq \min_{i \in L} \left\{ \frac{1}{\beta} - \|h_i(w^*)\| \right\} > 0.$$

Proof. By the non-expansiveness of $s(\cdot)$, for all i ,

$$(3.11) \quad \|\mathbf{w}_i^{k+1} - \mathbf{w}_i^*\| = \|s \circ h_i(w^k) - s \circ h_i(w^*)\| \leq \|h_i(w^k) - h_i(w^*)\|.$$

Suppose that at iteration k there exists at least one index $i \in L$ such that $\mathbf{w}_i^{k+1} = s \circ h_i(w^k) \neq 0$. Then, $\|h_i(w^*)\| < 1/\beta$, $\|h_i(w^k)\| \geq 1/\beta$ and $\mathbf{w}_i^* = s \circ h_i(w^*) = 0$. Therefore,

$$(3.12) \quad \begin{aligned} \|\mathbf{w}_i^{k+1} - \mathbf{w}_i^*\|^2 &= \|s \circ h_i(w^k)\|^2 = (\|h_i(w^k)\| - 1/\beta)^2 \\ &\leq \{\|h_i(w^k) - h_i(w^*)\| - (1/\beta - \|h_i(w^*)\|)\}^2 \\ &\leq \|h_i(w^k) - h_i(w^*)\|^2 - (1/\beta - \|h_i(w^*)\|)^2 \\ &\leq \|h_i(w^k) - h_i(w^*)\|^2 - \omega^2 \end{aligned}$$

where the first inequality is triangular inequality, the second follows from the fact that $\|h_i(w^k) - h_i(w^*)\| \geq 1/\beta - \|h_i(w^*)\| > 0$, and the last one uses the definition of ω in (3.10). Combining (3.11), (3.12) and the

non-expansiveness of $h(\cdot)$, we obtain

$$\begin{aligned}\|w^{k+1} - w^*\|^2 &= \sum_i \|\mathbf{w}_i^{k+1} - \mathbf{w}_i^*\|^2 \leq \sum_i \|h_i(w^k) - h_i(w^*)\|^2 - \omega^2 \\ &= \|h(w^k) - h(w^*)\|^2 - \omega^2 \leq \|w^k - w^*\|^2 - \omega^2.\end{aligned}$$

Therefore, the number of iterations k with $\mathbf{w}_i^{k+1} \neq 0$ does not exceed $\|w^0 - w^*\|^2/\omega^2$. \square

Given the finite convergence of $\mathbf{w}_i^k = \mathbf{w}_i^*$ for $i \in L$, we next show the linear convergence of u^k and \mathbf{w}_i^k for $i \in E$. Denote $w_E = ((w_1)_E; (w_2)_E)$, where $(w_1)_E$ and $(w_2)_E$ are the sub-vectors of w_1 and w_2 respectively with components $(w_1)_i, (w_2)_i, i \in E$.

THEOREM 3.6 (*q-linear convergence*). *Let M and T be defined as in (3.3) and $T_{EE} = [T_{i,j}]_{i,j \in E}$. Under Assumption 1, the sequence $\{(w^k, u^k)\}$ generated by Algorithm 1 satisfies*

1. $\|w_E^{k+1} - w_E^*\| \leq \sqrt{\rho((T^2)_{EE})} \|w_E^k - w_E^*\|;$
2. $\|D(u^{k+1} - u^*)\| \leq \sqrt{\rho((T^2)_{EE})} \|D(u^k - u^*)\|;$
3. $\|u^{k+1} - u^*\|_M \leq \sqrt{\rho(T_{EE})} \|u^k - u^*\|_M;$

for all k sufficiently large.

Proof. From (3.4 - 3.7) and the non-expansiveness of S , we get

$$(3.13) \quad u^{k+1} - u^* = M^{-1} D^T (w^{k+1} - w^*)$$

and

$$(3.14) \quad \|w^{k+1} - w^*\|^2 = \|S(D^{(1)}u^k; D^{(2)}u^k) - S(D^{(1)}u^*; D^{(2)}u^*)\|^2 \leq \|D(u^k - u^*)\|^2.$$

Combining the recursion (3.13), (3.14) and the definition of T , it holds

$$\|w^{k+1} - w^*\|^2 \leq \|T(w^k - w^*)\|^2.$$

Since we are interested in the asymptotic behavior of Algorithm 1, without loss of generality, hereafter we assume that $w_L^k \triangleq ((w_1^k)_L; (w_2^k)_L) = w_L^* = (0; 0)$. Therefore, the above inequality becomes

$$\|w_E^{k+1} - w_E^*\|^2 \leq (w_E^k - w_E^*)^T (T^2)_{EE} (w_E^k - w_E^*) \leq \rho((T^2)_{EE}) \|w_E^k - w_E^*\|^2.$$

Multiplying D on both sides of (3.13), from $w_L^k = w_L^* = 0$ and (3.14), we get

$$\|D(u^{k+1} - u^*)\|^2 \leq \rho((T^2)_{EE}) \|w^{k+1} - w^*\|^2 \leq \rho((T^2)_{EE}) \|D(u^k - u^*)\|^2.$$

The above two inequalities imply 1 and 2 of this theorem. From (3.13) and $w_L^k = w_L^* = 0$, we have

$$\|u^{k+1} - u^*\|_M^2 = (w^{k+1} - w^*)^T T (w^{k+1} - w^*) \leq \rho(T_{EE}) \|w^{k+1} - w^*\|^2.$$

Considering (3.14) and the definition of M , the above inequality becomes

$$\|u^{k+1} - u^*\|_M \leq \sqrt{\rho(T_{EE})} \|D(u^k - u^*)\| \leq \sqrt{\rho(T_{EE})} \|u^k - u^*\|_M,$$

namely, the last conclusion holds and thus u^k converges to u^* q -linearly. \square

Theorem 3.6 states that Algorithm 1 converges q -linearly with convergence rate depending on the spectral radii of the sub-matrices T_{EE} and $(T^2)_{EE}$ rather than on that of the whole matrix T . Since $\rho(T) \leq 1$ and T_{EE} is a minor of T , it holds that $\rho(T_{EE}) \leq \rho(T) \leq 1$. Similarly, $\rho((T^2)_{EE}) \leq \rho(T^2) \leq 1$. In the next section, we will demonstrate that in practice $\rho(T_{EE})$ can be much smaller than $\rho(T)$.

4. A Continuation Scheme. In this section, we develop a continuation scheme on the penalty parameter β based on our convergence results, and demonstrate its effectiveness.

It is easy to see from (3.3) that a smaller β generally gives a smaller $\rho(T)$. As is indicated in (3.9), a smaller β should also give a smaller set E , hence smaller $\rho(T_{EE})$ (this fact will be verified by our numerical experiments). These observations strongly suggest the use of a continuation scheme on β in which β is initially small and gradually increases to the final value. In this scheme, earlier subproblems corresponding to smaller β values can be solved quickly, and the later subproblems can also be solved relatively quickly by warm starting from the previous solutions.

To demonstrate the need for continuation and its effectiveness, we carried out a set of experiments on the 128×128 image Checkerboard, a blocky image available in MATLAB. Our experiments were done under MATLAB using its Image Processing Toolbox. A motion blurring kernel of motion distance “`len = 5`” and angle “`theta = 135`” was applied to the image with additive Gaussian noise of zero mean and standard deviation 10^{-3} through MATLAB functions “`fspecial`”, “`imfilter`” and “`imnoise`”. In our experiments, we set $\mu = 5 \times 10^4$.

In the first experiment, we compared the quantities $\rho(T)$ and $\rho(T_{EE})$ for different β values, both related to q -convergence rates. In addition, we also compared the theoretical upper bound, $N_0 \triangleq \|w^0 - w^*\|/\omega^2$, for finite convergence (see Theorem 3.5) with the observed iteration number, \hat{N}_0 , for finite convergence (i.e., after \hat{N}_0 iterations $w_i^k, i \in L$, remained zero until termination).

To do these calculations, we ran Algorithm 1 for three β values: $\beta = 2^4, 2^9$ and 2^{14} . For each β , an “exact” solution (u_β^*, w_β^*) of (1.3) was obtained using an extremely tight stopping tolerance $\epsilon = 10^{-15}$ in (2.10). Afterwards, we re-ran the algorithm for each β and stopped once $\|u^k - u_\beta^*\|/\|u_\beta^*\| < 1.2 \times 10^{-6}$. The calculated values of $\sqrt{\rho(T)}$, $\sqrt{\rho(T_{EE})}$, N_0 and \hat{N}_0 for the three β values are given in table 4.1.

TABLE 4.1
Comparison of theoretical convergence rates for different β

β	2^4	2^9	2^{14}
$\sqrt{\rho(T)}$	0.9999	0.9999	0.9999
$\sqrt{\rho(T_{EE})}$	0.5188	0.9955	0.9999
N_0	5.6850e+006	1.1990e+017	1.2103e+018
\hat{N}_0	5	139	3658

Data in Table 4.1 show that on the tested image (i) $\sqrt{\rho(T_{EE})}$ can be much smaller than $\sqrt{\rho(T)}$ when β is small and (ii) the theoretical worst-case upper bound for finite convergence can be vastly over-conservative in practice, especially when β is small.

In Figure 4.1, we plot the histories of the relative error, $\|u^k - u_\beta^*\|/\|u_\beta^*\|$, and the observed convergence factor, $\|u^{k+1} - u_\beta^*\|_M/\|u^k - u_\beta^*\|_M$, generated by Algorithm 1 for the three tested β values. The plots clearly illustrate that the convergence speed of Algorithm 1 is much faster for smaller β values. Specifically, when β increased from 2^4 to 2^{14} , the required iteration number for the given accuracy jumped from less than 10 to over 3000. Similarly, the observed q -convergence factor increased from around 0.2 to close to 1.

Our theoretical and numerical results strongly suggest the use of a continuation scheme on β . How

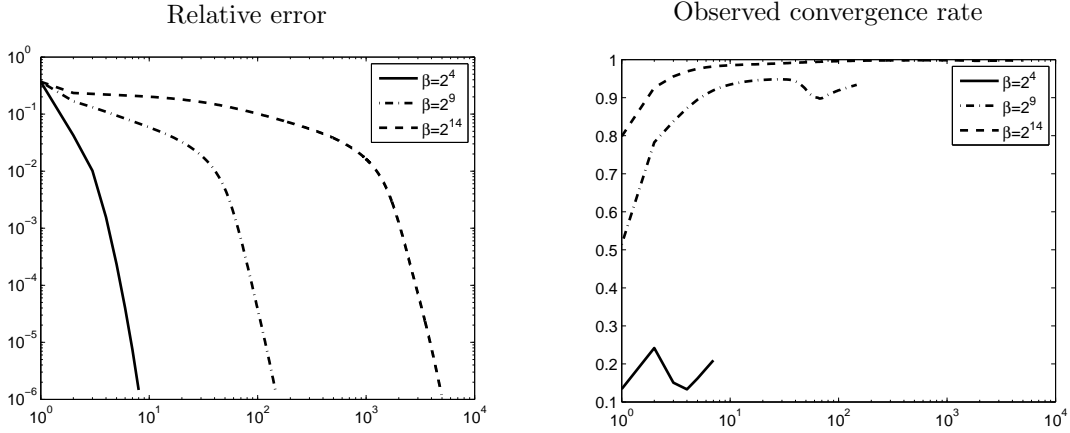


FIG. 4.1. Convergence behavior for $\beta = 2^4, 2^9$ and 2^{14} . The left plot is the relative error $e_k(\beta) = \|u^k - u_\beta^*\|/\|u_\beta^*\|$ and the right one is the observed q -linear convergence factor $q_k(\beta) = \|u^{k+1} - u_\beta^*\|_M/\|u^k - u_\beta^*\|_M$. In each plot, the horizontal axis represents the number of iterations.

to do this continuation most efficiently should be an interesting research issue by its own right, but will not be studied in depth in the current paper. Instead, we implement a very basic version, likely still far from optimal, to test the viability of our proposed framework. We call the resulting algorithm “Fast Total Variation de-convolution” or FTVd, which is given below.

ALGORITHM 2 (FTVd). *Input* $f, K, \mu > 0, \beta_0 > 0$ and $\beta_{\max} > \beta_0$.

Initialize $u = f, u_p = 0, \beta = \beta_0$ and $\epsilon > 0$.

While $\beta \leq \beta_{\max}$, **Do**

1) Run Algorithm 1 until (2.10) is met.

2) $\beta \leftarrow 2 * \beta$.

End Do

One can modify the above framework to make it more flexible, though as it is the above basic implementation already works surprisingly well. For example, one could adaptively increase β and choose ϵ from one outer iteration to another (or use another stopping criterion for Algorithm 1).

We tested the effectiveness of FTVd for $\beta_0 = 1$ and $\beta_{\max} = 2^{14}$ and present the results in Figure 4.2 in comparison to the results without doing continuation. From this comparison, it is abundantly clear that the algorithm would not be practically effective without continuation.

5. Numerical Experiments. In this section, we present detailed numerical results comparing FTVd to the Lagged Diffusivity (LD) algorithm [42, 10], a state-of-the-art method for solving the isotropic TV deblurring model (1.2), as well as to some non-TV deblurring algorithms.

5.1. Overall assessment of several algorithms. We tested several popular algorithms for solving the TV deblurring model (1.2). Based on our experience, we have reached the following overall impression.

The artificial time-marching algorithm used by Rudin and Osher in [38, 39] is easy to implement, but usually takes a large number of iterations, each with a small step size governed by the CFL condition, to reach a modest accuracy. On the other hand, the second order cone programming (SOCP) approach [19] recovers solutions with a high accuracy in a small number of iterations (typically 30-50), but takes much

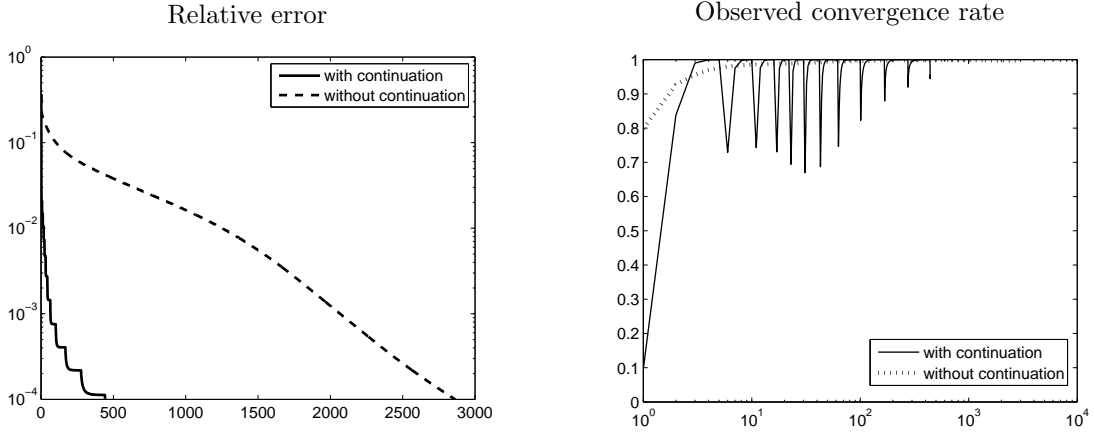


FIG. 4.2. Continuation vs. no continuation: u^* is an “exact” solution corresponding to $\beta = 2^{14}$. The left plot is the relative error $e_k = \|u^k - u^*\| / \|u^*\|$ and the right one is the observed q -linear convergence factor $q_k = \|u^{k+1} - u^*\|_M / \|u^k - u^*\|_M$. In each plot, the horizontal axis represents the number of iterations.

more running time and memory in each iteration even on medium-sized images. A comprehensive comparison between the artificial time-marching algorithm and the SOCP approach is presented in [19]. We did not test the interior-point primal-dual implicit quadratic methods presented in [9, 3, 30, 8, 30], but expect their performance to be similar to SOCP because they also require solving large systems of linear equations at each iteration. We also tested the recently proposed iteratively re-weighted least squares algorithm [37] and obtained consistent speed and accuracy as reported by the authors. Specifically, it was slower than the Lagged Diffusivity (LD) method devised by Vogel and Oman [42, 10] for solving the TV/ L^2 model (1.2). Therefore, among all tested algorithms, LD is the most efficient for solving (1.2). Clearly, there are other deblurring algorithms not included in our discussion, but a more exhaustive comparison is beyond the scope of this work.

5.2. Test images and test platforms. We used two images, Lena and Man in our experiments, see Figure 5.1. Image Lena is a good test image because it has a nice mixture of detail, flat regions, shading area and texture. Image Man also consists of complex components in different scales, with different patterns and under inhomogeneous illuminations. Both images are suited for our experiments.

We tested several kinds of blurring kernels including Gaussian, average and motion, and found that the running times of FTVd and LD remained essentially constant for different blurring kernels as long as other conditions (e.g., kernel size or image size) were the same. Therefore without loss of generality, we only present results using “Gaussian” blurring kernel with different kernel sizes. In all tests, the additive noise used was Gaussian noise with zero mean and standard deviation 10^{-3} . This level of noise is substantial in the context of deblurring. Detailed information of our experiment set-up is summarized in Table 5.1.

FTVd was implemented in MATLAB. All blurring effects were generated using the MATLAB function “`imfilter`” with periodic boundary conditions. For LD, we used an efficient C++ code NUMIPAD [36]. NUMIPAD uses Dirichlet boundary conditions for TV calculation (i.e. areas outside of the image boundary are treated as 0) instead of periodic boundary conditions. This difference caused the two codes to generate images with minor differences near boundaries, but should not noticeably affect their running times.



FIG. 5.1. *Test images: Lena (left) and Man (right).*

TABLE 5.1
Information on test images and blurring setting

Test No.	Image	Size	Blurring Type	Blurring Kernel Parameters
1.	Lena	512×512	Gaussian	hsize={3, 5, \dots , 19, 21}, $\sigma = 10$
2.	Man	1024×1024	Gaussian	hsize={3, 5, \dots , 19, 21}, $\sigma = 10$

All comparisons between FTVd and LD were performed under GNU/Linux Release 2.6.9-55.0.2 and MATLAB v7.2 (R2006a) running on a DELL Optiplex GX620 with Dual Intel Pentium D CPU 3.20GHz (only one processor was used by MATLAB) and 3 GB of memory. The C++ code of LD was a part of the library package NUMIPAD [36] and was compiled with gcc v3.4.6. Since the code of ForWaRD had compiling problems on our Linux workstation, we chose to compare FTVd with ForWaRD, and also MATLAB functions “deconvnr” and “deconvreg” under Windows XP and MATLAB v7.5 (R2007b) running on a Lenovo laptop with an Intel Core 2 Duo CPU at 2 GHz and 2 GB of memory.

5.3. Parameter values. In model (1.2), the parameter μ controls the amount of penalty applied to the squared L^2 -distance between Ku and f . According to (1.1), an appropriate μ should give a solution u of (1.2) satisfying $\|Ku - f\| \approx \|\omega\|$. In our experiments, we used $\mu = 0.05/\sigma^2$, where σ is the standard deviation of the additive Gaussian noise ω in (1.1). This formula is based on the observation that μ should be inversely proportional to the noise variance, while the constant 0.05 was determined empirically so that the restored images had reasonable signal-to-noise ratios (will be defined latter) and relative errors. More practical techniques exist for choosing μ , often by testing on a small window of image. The reader interested in this topic is referred to [40]. The issue of how to optimally select μ is important, but outside of the scope of this work.

As is usually done, we measure the quality of restoration by the signal-to-noise ratio (SNR), defined by

$$\text{SNR} \triangleq 10 * \log_{10} \frac{\|u^0 - \tilde{u}\|^2}{\|u^0 - u\|^2},$$

where u^0 is the original image and \tilde{u} is the mean intensity value of u^0 . Generally, the quality of the restored image is expected to increase as β increases (hence, (1.3) becomes a closer approximation to (1.2)). In practice, we observed that the SNR value of recovered images from (1.3) stabilized once β reached a reasonably large value. To see this, we plot the SNR values of restored images corresponding to $\beta = 2^0, 2^1, \dots, 2^{18}$ in Figure 5.2. In this experiment, ‘‘Gaussian’’ out-of-focus blurring was applied to ‘‘Lena’’ with blurring size `hsize = 11` and standard deviation $\sigma = 5$, and the ‘‘motion’’ blurring applied to ‘‘Man’’ with motion distance `len = 21` and `theta = 135`.

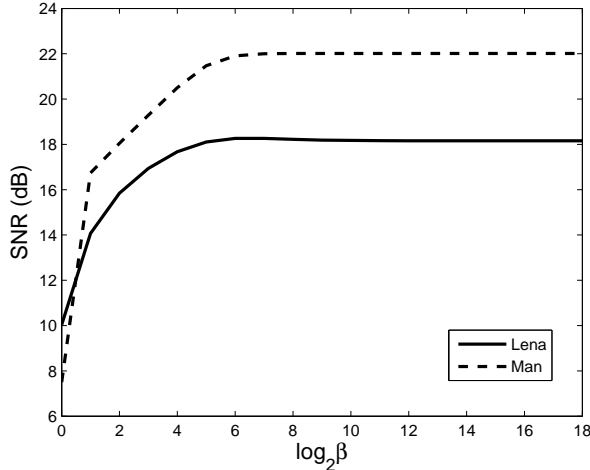


FIG. 5.2. SNRs of images recovered from (1.3) for different β .

As can be seen from Figure 5.2, the SNR values on both images essentially remain constant for $\beta \geq 2^7$. This suggests that β need not to be excessively large from a practical point of view. In our numerical experiments comparing FTVd with LD, we set $\beta_0 = 1$ and $\beta_{\max} = 2^7$ in Algorithm 1. For each β , the inner iteration was stopped once (2.10) was satisfied with $\epsilon = 0.05$.

5.4. Comparison of FTVd and LD. In this subsection, we compare FTVd with LD in terms of both convergence speed and quality of recovery. As mentioned in subsection 5.2, the two methods use different boundary conditions which causes minor differences. The blurry images with the circular boundary condition usually have slightly higher SNRs than those blurred with the Neumann boundary condition, and so do the restored images. To have a more fair comparison, we measure the restoration quality by the so-called Improvement SNR (or ISNR) defined as

$$\text{ISNR} \triangleq 10 * \log_{10} \frac{\|f - u^0\|^2}{\|u - u^0\|^2},$$

where u^0 , u and f are the original, the restored, and the observed images, respectively. This quantity measures the quality of a restored image u relative to the blurry and noisy observation f .

The ISNRs of the recovered images computed by FTVd and LD methods for the two test images are given in the left plot of Figure 5.3. We observe that FTVd produced slightly lower ISNRs when `hsize` is

small and slightly higher ISNRs when `hsize` is large. However, since both FTVd and LD solve the same model (boundary conditions aside), the restored images have little visible differences. Therefore, we will not display the restored images here.

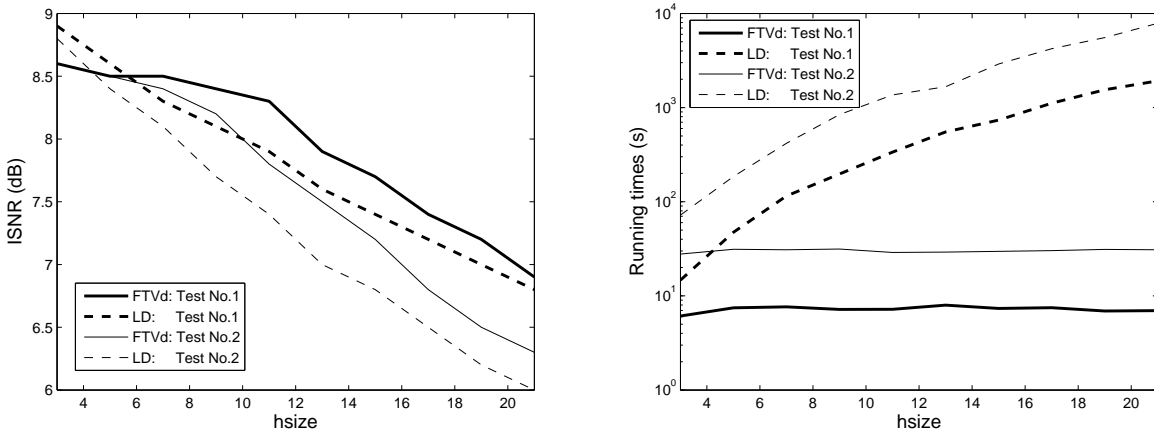


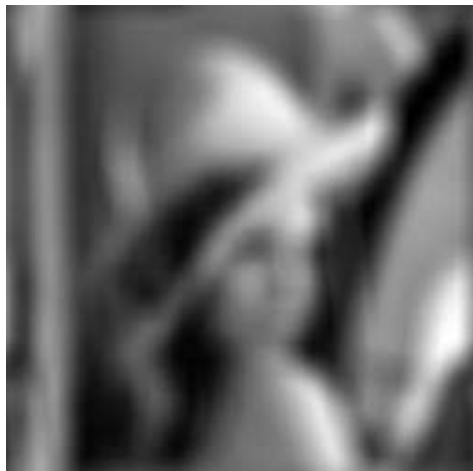
FIG. 5.3. ISNRs (left) and CPU time (right) comparisons for FTVd and LD methods. In both figures, the horizontal axis represents the size of blurring kernel.

The right plot in Figure 5.3 shows the comparison of running time between FTVd and LD. While generating the restored images with similar qualities, our MATLAB code FTVd is much faster than the C++ code for LD. The running time of FTVd essentially remains constant for different values of `hsize`, and increases only moderately when image size is doubled. However, the running time for LD increases dramatically with the increase in both the image size and the blurring kernel size. As `hsize` becomes relatively large, the CPU times consumed by FTVd and LD become orders of magnitude apart. The reason is clear: larger `hsize` makes the matrix K denser and more ill-conditioned; hence, solving a linear system involving K become more difficult for the pre-conditioned conjugate gradient (CG) method used in NUMIPAD [36]. On the other hand, the increase of kernel size does not noticeably increase the cost of doing FFT on the function represented by K .

5.5. Observations on FTVd’s behavior and a note. As has been mentioned, the per-iteration computation of FTVd is dominated by three FFTs, each at the cost of $O(n^2 \log(n))$. The question is how many iterations are needed in general for FTVd to attain a required accuracy for images of different sizes. In our experiments, the number of total inner iterations taken by the basic version of FTVd, with the given “default” parameters, is almost always around 12. These total inner iteration numbers are quite reasonable, given that there were 8 outer-iterations corresponding to the β -sequence $\{2^0, 2^1, \dots, 2^7\}$. For each $\beta > 1$, since FTVd has already obtained a good starting point, on average it only takes 1 or 2 inner iterations to reach the prescribed accuracy. Given that three FFTs are required at each inner iteration, the total numbers of FFTs taken by FTVd is around 40 for all tests. Upon examining detailed timing information, we observed that over two thirds of the CPU time was spent on the FFT and related calculations in (2.4). Finally, FTVd is numerically stable and insensitive to ill-conditioning, apparently because it does not require any matrix operations.

After we finished the writing of this paper, we found a newly released paper by Huang, Ng and Wen

Blurry&Noisy. SNR: 5.19dB



ForWaRD: SNR: 12.46dB, t = 4.88s



FTVd: $\beta = 2^5$, SNR: 12.58dB, t = 1.83s



FTVd: $\beta = 2^7$, SNR: 13.11dB, t = 14.10s



deconvwnr: SNR: 11.51dB, t = 0.05s



deconvreg: SNR: 11.20dB, t = 0.34s



FIG. 5.4. The first row contains the blurry and noisy image and the image recovered by ForWaRD; the second row contains the results of FTVd (left: $\beta = 2^5$, $\epsilon = 5 \times 10^{-2}$; right: $\beta = 2^7$, $\epsilon = 2 \times 10^{-3}$); and the third row is recovered by deconvwnr (left) and deconvreg (right)

[23] which solves the following approximation to the TV/ L^2 model (after necessary notational changes for consistency)

$$(5.1) \quad \min_{u,v} \sum_i \|D_i v\| + \alpha \|v - u\|^2 + \frac{\mu}{2} \|Ku - f\|^2,$$

by alternating minimization. For a fixed v , the minimization with respect to u involves 2 FFTs, one less than what is required by the corresponding subproblem in our method. However, for a fixed u , the minimization with respect to v is nothing but a TV denoising problem which does not have a closed-form solution with linear complexity as our corresponding subproblem has (see formula (2.2)). In [23], the TV denoising problem is solved iteratively by Chambolle’s projection algorithm [6]. While the per-iteration computational complexity of our method is dominated by 3 FFTs, that of [23] is dominated by the cost of solving a TV denoising problem in addition to 2 FFTs. According to the reported numerical results in [23], their algorithm appears to require at least as many outer iterations as ours.

5.6. Comparison of FTVd with other methods. We also conducted tests to show how FTVd compares with some other state-of-the-art, non-total-variation methods. A hybrid Fourier-wavelet regularized deconvolution (ForWaRD) algorithm by Neelanmani, Choi and Baraniuk [32] uses shrinkage in both Fourier and wavelet domains to recover images while preserving edges and removing noise. Their code was written in the C programming language with a MATLAB interface¹. We compared the quality of images recovered by FTVd, ForWaRD and two MATLAB deblurring functions “`deconvwnr`” (Wiener filter) and “`deconvreg`” (Laplacian low-pass filter). These methods are all based on distinct models

In this experiment, we used the image Lena, which was blurred with a “Gaussian” kernel of `hsize = 21` and standard deviation $\sigma = 11$. We ran FTVd twice, once with $\beta = 2^5$ and tolerance $\epsilon = 0.05$ in (2.10), and another with $\beta = 2^7$ and $\epsilon = 0.002$. The blurry and noisy and the restored images are presented in Figure 5.4. SNRs of the restored images and running times of different algorithms are also given. As can be seen from Figure 5.4, FTVd is comparable with ForWaRD in speed and attains a better image quality. The MATLAB functions are faster as they solve simpler models, but generate images of lower quality. Specifically, the artifacts of multiple rings or ripples are clearly more visible in the restored images by the three methods than in the ones by FTVd. By visually comparing the restored images, we give preference to the results of FTVd.

5.7. Summary of numerical results. First, FTVd can be orders of magnitude faster than the Lagged Diffusivity (LD) method. The performance gap between them will continue to grow as the image and blurring kernel sizes increase. Since LD compared favorably to many other existing algorithms for solving (1.2) with isotropic TV, the lead of FTVd obviously extends over to those algorithms. Secondly, compared to some other non-total-variation deblurring algorithms like ForWaRD and the MATLAB deblurring functions, FTVd generally produces images of higher quality in our experiments on numerous natural images. Under the “default” parameter setting, the speed of FTVd is not as fast as the MATLAB functions “`deconvwnr`” and “`deconvreg`”. However, with a less restrictive stopping tolerance and a smaller β_{\max} , the running time of FTVd can be faster than that of “`deconvreg`” on large images and close to that of “`deconvwnr`” while still producing images of higher quality.

¹The code ForWaRD can be downloaded at <http://www.dsp.rice.edu/software/ward.shtml>.

6. Extensions. There are some obvious extensions in which one can apply the variable-splitting and alternating-minimization methodology to solving other problems. Here we mention a few.

Color or other multi-channel image reconstruction with TV regularization. This extension is practically important, as straightforward as it may be, because it enables multi-channel TV image reconstruction problems to be solved at a speed not seen before. We are currently working on this extension and will report our findings in a forthcoming paper. In addition, we can also extend this methodology to models where the fidelity term $\|Ku - f\|$ is not measured in the 2-norm but the 1-norm in order to handle impulsive noise (such as “salt and pepper”) other than white noise.

The use of higher-order finite difference operators. In this case, we may have models in the form of

$$(6.1) \quad \min_u \sum_i \|D_i^{(p)}u\| + \frac{\mu}{2}\|Ku - f\|^2,$$

for some $p > 2$, which can be approximated by

$$(6.2) \quad \min_{\mathbf{w}, u} \sum_i \left(\|\mathbf{w}_i\| + \frac{\beta}{2}\|\mathbf{w}_i - D_i^{(p)}u\|^2 \right) + \frac{\mu}{2}\|Ku - f\|^2.$$

As long as the difference operators $D_i^{(p)}$ are linear and shift-invariant, minimization with respect to u is as simple as a few FFTs for a fixed \mathbf{w} ; while minimization with respect to \mathbf{w} can be done by shrinkage in an appropriate dimension (depending on the order of the finite difference involved). Higher-order derivatives of u are used in [27, 44], for example, to reduce staircasing effects.

Operator K is a partial Fourier transform matrix. In this case, $Ku = P\mathcal{F}(u)$ where $P \in \mathbb{R}^{m \times n^2}$ is a selection matrix consisting of $m < n^2$ rows of the identity matrix. This case occurs, for example, in compressive sensing [4, 16] where a subset of noisy Fourier coefficients, f , is used to recover a signal u^0 with a sparse gradient field via solving the model

$$(6.3) \quad \min_u \sum_i \|D_i u\| + \frac{\mu}{2}\|P\mathcal{F}(u) - f\|^2.$$

It is easy to verify that Algorithm 1 can be applied to the above problem with minor modifications to formula (2.4), including replacing $\mathcal{F}(K)$ by a section vector $p \in \mathbb{R}^{n^2}$ whose entries are 1 corresponding to rows selected by P and zero otherwise. We have applied this approach to solving simulation problems in compressive magnetic resonance (MR) imaging with promising results.

Multiple regularization terms. Such examples arise in both image processing (e.g., [28, 8]) and compressive MR imaging (e.g., [22, 26, 29]). For example, the model

$$(6.4) \quad \min_u \sum_i \|D_i u\| + \alpha\|\Phi u\|_1 + \frac{\mu}{2}\|P\mathcal{F}(u) - f\|^2,$$

where Φ is an orthonormal matrix, can be approximated by

$$(6.5) \quad \min_{u, v, \mathbf{w}} \left(\sum_i \|\mathbf{w}_i\| + \frac{\beta}{2}\|\mathbf{w}_i - D_i u\|^2 \right) + \alpha \left(\|v\|_1 + \frac{\gamma}{2}\|v - \Phi u\|^2 \right) + \frac{\mu}{2}\|P\mathcal{F}(u) - f\|^2.$$

In an alternating minimization setting, w can be solved by (2.2) as before, v can be solved by one-dimensional shrinkage, and u can be solved by 4 FFTs (2 for w_1 and w_2 , one for $\Phi^T v$ noting that $\|v - \Phi u\| = \|\Phi^T v - u\|$,

and one inverse FFT). In general, one needs to introduce an auxiliary variable for each regularization term and manages to obtain easily solvable subproblems.

Finally, we mention that even in the case where the quadratic part in (1.3) cannot be minimized with respect to u via fast transforms like in formula (2.4), model (1.3) may still lead to an efficient algorithm if the quadratic in u can be effectively decreased by a few iterations of an iterative method such as a pre-conditioned CG method. This topic is of interest for further investigation.

7. Concluding remarks. We proposed, analyzed and tested a new algorithm FTVd for solving the TV/ L^2 model (1.2). Our approximation model (1.3) was derived from the classic quadratic penalty method first proposed by Courant in 1943, but also falls into the more recent half-quadratic class for image processing after our extensions. In fact, it can be regarded as a long missing, isotropic TV model in the half-quadratic class whose quadratic portion has a block circulant Hessian, thus allowing fast transforms to be utilized in quadratic minimization.

We established strong convergence results for our algorithm and validated a continuation scheme. Our numerical results convincingly show that FTVd can be orders of magnitude faster than the Lagged Diffusivity algorithm. In particular, our results demonstrate that the TV/ L^2 model can be solved just as quickly as some other more widely used models, putting it on an equal footing with others in terms of solution speed.

The results of this paper can be extended to a number of models involving TV regularization. More studies are under way to further explore the potential of the proposed approach.

Acknowledgments. We are grateful to Paul Rodríguez and Brendt Wohlberg at the Los Alamos National Lab for valuable discussions and making their algorithm library NUMIPAD [36] available to us. The work of Y. Wang and W. Yin has been supported by the latter author's internal faculty research grant from the Dean of Engineering at Rice University. 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-0405831.

REFERENCES

- [1] R. ACAR AND C. R. VOGEL, *Analysis of total variation penalty methods*, Inv. Probl., 10 (1994), pp. 1217–1229.
- [2] M. ALLAIN, J. IDIER, AND Y. GOUSSARD, *On global and local convergence of half-quadratic algorithm*, IEEE Transactions on Image Processing, 15 (2006), pp. 1130–1142.
- [3] P. BLOMGREN, T. F. CHAN, AND C. K. WONG, *Total variation image restoration: Numerical methods and extensions*, in Proceedings of IEEE international conference on image processing, vol. 3, 1997, pp. 384–387.
- [4] E. CANDÈS, J. ROMBERG, AND T. TAO, *Stable signal recovery from incomplete and inaccurate information*, Communications on Pure and Applied Mathematics, 2005 (2005), pp. 1207–1233.
- [5] A. S. CARASSO, *Linear and nonlinear image deblurring: A documented study*, SIAM J. Numer. Anal., 36 (1999), pp. 1659–1689.
- [6] A. CHAMBOLLE, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision., 20 (2004), pp. 89–97.
- [7] A. CHAMBOLLE AND P. L. LIONS, *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (1997), pp. 167–188.
- [8] T. F. CHAN, S. ESEDOGLU, F. PARK, AND A. YIP, *Recent developments in total variation image restoration*, CAM Report 05-01, Department of Mathematics, UCLA, 2004.

- [9] T. F. CHAN, G. H. GOLUB, AND P. MULET, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput., 20 (1999), pp. 1964–1977.
- [10] T. F. CHAN AND P. MULET, *On the convergence of the lagged diffusivity fixed point method in total variation image restoration*, SIAM J. Numer. Anal., 36 (1999), pp. 354–367.
- [11] T. F. CHAN AND J. SHEN, *Theory and computation of variational image deblurring*, IMS Lecture Notes, (2006).
- [12] P. CHARBONNIER, L. BLANC-FÉRAUD, G. AUBERT, AND M. BARLAUD, *Deterministic edge preserving regularization in computed imaging*, IEEE Transactions on Image Processing, 6 (1997), pp. 298–311.
- [13] R. COURANT, *Variational methods for the solution of problems with equilibrium and vibration*, Bull. Amer. Math. Soc., 49 (1943), pp. 1–23.
- [14] G. DEMOMENT, *Image reconstruction and restoration: Overview of common estimation structures and problems*, IEEE Trans. Acoust., Speech, Signal Process., 37 (1989), pp. 2024–2036.
- [15] D. C. DOBSON AND F. SANTOSA, *Recovery of blocky images from noisy and blurred data*, SIAM J. Appl. Math., 56 (1996), pp. 1181–1198.
- [16] D. DONOHO, *Compressed sensing*, IEEE Transactions on Information Theory, 52 (2006), pp. 1289–1306.
- [17] D. GEMAN AND G. REYNOLDS, *Constrained restoration and the recovery of discontinuities*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 (1992), pp. 367–383.
- [18] D. GEMAN AND C. YANG, *Nonlinear image recovery with half-quadratic regularization*, IEEE Transactions on Image Processing, 4 (1995), pp. 932–946.
- [19] D. GOLDFORB AND W. YIN, *Second-order cone programming methods for total variation-based image restoration*, SIAM J. Sci. Comput., 27 (2005), pp. 622–645.
- [20] T. GOLDSTEIN AND S. OSHER, *The split Bregman algorithm for L1 regularized problems*, UCLA CAM Report 08-29, (2008).
- [21] R. GONZALEZ AND R. WOODS, *Digital Image Processing*, Addison-Wesley, 1992.
- [22] L. HE, T.-C. CHANG, S. OSHER, T. FANG, AND P. SPEIER, *MR image reconstruction by using the iterative refinement method and nonlinear inverse scale space methods*, UCLA CAM Report 06-35, (2006).
- [23] Y. HUANG, M. K. NG, AND Y. WEN, *A fast total variation minimization method for image restoration*, SIAM Journal on Multiscale Modeling and Simulation, to appear.
- [24] J. IDIER, *Convex half quadratic criteria and interacting auxiliary variables for image restoration*, IEEE Transactions on Image Processing, 10 (2001), pp. 1001–1009.
- [25] A. K. KATSAGGELOS, *Digital Image Restoration*, Springer-Verlag, 1991.
- [26] M. LUSTIG, D. DONOHO, AND J. PAULY, *Sparse MRI: The application of compressed sensing for rapid MR imaging*, Magnetic Resonance in Medicine, in press (2007).
- [27] M. LYSAKER, A. LUNDERVOLD, AND X. C. TAI, *Noise removal using fourth-order partial differential equations with applications to medical magnetic resonance images in space and time*, IEEE Transactions on Image Processing, 12 (2002), pp. 1579–1590.
- [28] O. M. LYSAKER AND X.-C. TAI, *Iterative image restoration combining total variation minimization and a second-order functional*, International Journal of Computer Vision, 66 (2006), pp. 5–18.
- [29] S. MA, W. YIN, Y. ZHANG, AND A. CHAKRABORTY, *Compressed MR imaging based on wavelets and total variation*, In proceedings of CVPR’08, (2008).
- [30] A. MARQUINA AND S. OSHER, *Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal*, SIAM J. Sci. Comput., 22 (2001), pp. 387–405.
- [31] R. NEELAMANI, H. CHOI, AND R. G. BARANIUK, *Wavelet-based deconvolution for ill-conditioned systems*, in Proc. IEEE ICASSP, vol. 6, Mar. 1999, pp. 3241–3244.
- [32] ———, *ForWaRD: Fourier-Wavelet regularized deconvolution for ill-conditioned systems*, IEEE Trans. signal proc., 52 (2004), pp. 418–433.
- [33] 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.
- [34] M. NIKOLOVA AND M. K. NG, *Analysis of half-quadratic minimization methods for signal and image recovery*, SIAM J. Sci. Comput., 27 (2005), pp. 937–966.
- [35] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer, 2000.

- [36] P. RODRÍGUEZ AND B. WOHLBERG, *Numerical methods for inverse problems and adaptive decomposition (NUMIPAD)*.
<http://numipad.sourceforge.net/>.
- [37] ———, *An iteratively weighted norm algorithm for total variation regularization*, To appear in IEEE Signal Processing Letters, (2007).
- [38] L. RUDIN AND S. OSHER, *Total variation based image restoration with free local constraints*, Proc. 1st IEEE ICIP, 1 (1994), pp. 31–35.
- [39] L. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica D, (1992), pp. 259–268.
- [40] D. STRONG AND T. F. CHAN, *Edge-preserving and scale-dependent properties of total variation regularization*, Inverse Problems, 19 (2003), pp. 165–187.
- [41] A. TIKHONOV AND V. ARSEININ, *Solution of Ill-Posed problems*, Winston, Washington, DC, 1977.
- [42] C. R. VOGEL AND M. E. OMAN, *Iterative methods for total variation denoising*, SIAM J. Sci. Comput., 17 (1996), pp. 227–238.
- [43] Y. WANG, W. YIN, AND Y. ZHANG, *A fast algorithm for image deblurring with total variation regularization*. CAAM Technical Report 07-10, Rice University, June 2007.
- [44] Y. L. YOU AND M. KAVEH, *Fourth-order partial differential equations for noise removal*, IEEE Tran. on image processing, 9 (2000), pp. 1723–1730.