Efficient minimization methods of mixed $\ell 2$-$\ell 1$ and $\ell 2$-$\ell 1$ norms for image restoration

Fu, Haoying; Ng, Michael K.; Nikolova, Mila; Barlow, Jesse L.

Published in:
SIAM Journal on Scientific Computing

DOI:
10.1137/040615079

Published: 25/07/2006

Document Version:
Publisher's PDF, also known as Version of record

SIAM J. SCI. COMPUT.
Vol. 27, No. 6, pp. 1881–1902

EFFICIENT MINIMIZATION METHODS OF MIXED $\ell 2$-$\ell 1$ AND $\ell 1$-$\ell 1$ NORMS FOR IMAGE RESTORATION*

HAOYING FU†, MICHAEL K. NG‡, MILA NIKOLOVA§, AND JESSE L. BARLOW†

Abstract. Image restoration problems are often solved by finding the minimizer of a suitable objective function. Usually this function consists of a data-fitting term and a regularization term. For the least squares solution, both the data-fitting and the regularization terms are in the $\ell 2$ norm. In this paper, we consider the least absolute deviation (LAD) solution and the least mixed norm (LMN) solution. For the LAD solution, both the data-fitting and the regularization terms are in the $\ell 1$ norm. For the LMN solution, the regularization term is in the $\ell 1$ norm but the data-fitting term is in the $\ell 2$ norm. Since images often have nonnegative intensity values, the proposed algorithms provide the option of taking into account the nonnegativity constraint. The LMN and LAD solutions are formulated as the solution to a linear or quadratic programming problem which is solved by interior point methods. At each iteration of the interior point method, a structured linear system must be solved. The preconditioned conjugate gradient method with factorized sparse inverse preconditioners is employed to solve such structured inner systems. Experimental results are used to demonstrate the effectiveness of our approach. We also show the quality of the restored images, using the minimization of mixed $\ell 2$-$\ell 1$ and $\ell 1$-$\ell 1$ norms, is better than that using only the $\ell 2$ norm.

Key words. least squares, least absolute deviation, interior point method, linear programming, quadratic programming, preconditioner

AMS subject classifications. 65C20, 65F10

DOI. 10.1137/040615079

1. Introduction.

The problem of image restoration is considered. The observed image is the convolution of a shift invariant blurring function with the true image plus some additive noise. Let $f(x, y)$ be the original scene, $g(x, y)$ the observed scene, and $h(x, y)$ the blurring function. The image formation process can be modeled as follows:

$$g(x, y) = h(x, y) \star f(x, y) + w(x, y). \tag{1.1}$$

Here $w(x, y)$ is the additive noise and $\star$ denotes two-dimensional convolution. Let $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{w}$ be the discretized original scene, observed scene, and additive noise, respectively. Let H be the corresponding blurring matrix of appropriate size built according to the discretized point spread function $\mathbf{h}$. Then the discretized image formation process can be put into matrix-vector form:

$$\mathbf{g} = H\mathbf{f} + \mathbf{w}. \tag{1.2}$$

Assuming that the discretized scenes have $m \times n$ pixels, then $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{w}$ are vectors of length mn, and H is a matrix of $mn \times mn$. We remark that H is a matrix of block

*Received by the editors September 14, 2004; accepted for publication (in revised form) March 29, 2005; published electronically February 3, 2006.
http://www.siam.org/journals/sisc/27-6/61507.html

†Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802 (hfu@cse.psu.edu, barlow@cse.psu.edu). Research for this work was done while the first author was visiting the Department of Mathematics, The University of Hong Kong.

‡Department of Mathematics, The University of Hong Kong, Hong Kong (mng@maths.hku.hk). The research of this author was supported in part by Hong Kong Research Grants Council grants HKU 7130/02P, HKU 7046/03P, and HKU 7035/05P.

§Centre de Mathématiques et de Leurs Applications, ENS de Cachan, 61 av. President Wilson, 94235 Cachan Cedex, France (nikolova@cmla.ens-cachan.fr).

Toeplitz with Toeplitz blocks (BTTB) when zero boundary conditions are applied, and block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel blocks (BTHTHB) when Neumann boundary conditions are used [36].

Image restoration problems tend to be very ill-conditioned. Directly solving (1.2) will yield a solution that is extremely sensitive to noise; therefore, regularization methods are needed to stabilize the solution. For example, the linear least squares problem with Tikhonov's regularization [48] takes the following form:

$$\min_{\mathbf{f}} \|\mathbf{g} - H\mathbf{f}\|_2^2 + \alpha \|R\mathbf{f}\|_2^2. \tag{1.3}$$

In this optimization problem, the second term is a regularization term that measures the "irregularity" of the solution. We call R the regularization operator and α the regularization parameter. Very often R is chosen to be the difference operator, for example, the first order or second order finite difference operator.

Assuming the additive noise is white Gaussian and the prior distribution of $\mathbf{f}$ is also Gaussian, and that exact information of the blurring matrix is available, the solution of (1.3) can be interpreted as the "maximum a posteriori (MAP) estimator" of the original scene; see [6], for instance. However, due to the presence of edges, the prior distribution of an image rarely satisfies the Gaussian assumption well. In many cases, the additive noise does not satisfy the Gaussian assumption either; for instance, the noise may follow a Laplace distribution [6]. In the literature, there has been a growing interest in using the $\ell 1$ norm for parameter estimation [4, 6, 5, 26, 9, 10, 44] and for image restoration [3, 12, 21, 28, 35, 38, 39, 25]. The advantage of using the $\ell 1$ norm is that the solution is more robust than when using the $\ell 2$ norm in statistical estimation problems. In particular, a small number of outliers have less influence on the solution; see, for instance, [9, 38].

Since edges in an image lead to outliers in the regularization term, it is natural to consider using the $\ell 1$ norm for this term:

$$\min_{\mathbf{f}} \|\mathbf{g} - H\mathbf{f}\|_2^2 + \alpha \|R\mathbf{f}\|_1. \tag{1.4}$$

We call the solution to (1.4) the least mixed norm (LMN) solution. It is worth pointing out that model (1.4) is closely related to the total variation regularization in image restoration. In the one-dimensional case, if R is chosen to be the first order difference operator, then $\|R\mathbf{f}\|_1$ is the discrete version of the total variation norm. Li and Santosa used a piecewise linear $\ell 1$ function as a measure of total variation in [29]. For details about total variation regularization, we refer the reader to [43, 29, 13, 24, 11, 23].

If the additive noise does not satisfy the Gaussian assumption either, we consider using the $\ell 1$ norm for both the data-fitting and the regularization terms:

$$\min_{\mathbf{f}} \|\mathbf{g} - H\mathbf{f}\|_1 + \alpha \|R\mathbf{f}\|_1. \tag{1.5}$$

We call the solution to (1.5) the least absolute deviation (LAD) solution. Recently, minimizers of cost-functions involving $\ell 1$ data-fidelity have been studied by Nikolova [38, 39], Chan and Esedoglu [12], and Kärkkäinen, Kunisch, and Majava [25]. Applications to the processing of outliers and the removal of impulse noise are considered in denoising problems [38, 39, 25].

Another assumption that must be satisfied in order for (1.3) to yield the MAP estimator and which is often violated is that the exact information of the blurring matrix H is available. Many papers (for instance, [34, 37, 42, 20, 33]) address the issue

of inexact H but still assume white Gaussian noise and Gaussian prior distribution of $\mathbf{f}$. In this work, we assume that the exact information for H is available and focus on the issues of non-Gaussian noise and non-Gaussian prior distribution.

Since the objective functions are nonsmooth, problems (1.4) and (1.5) are difficult to solve [6, 9]. For instance, Mammone and Eichmann [32] solved an $\ell 1$ approximation problem with a nonnegative solution using linear programming techniques, and Abdelmalek and Kasvand [2, 1] solved the problem as a quadratic programming problem whose solution is nonnegative and bounded. However, they studied the algebraic image restoration problem in one dimension only, for example, separable blurring functions. Pruessner and O'Leary [42] considered minimization of total least $\ell 1$ norm by using linear programming for image restoration. However, since they simply use the MATLAB [19] function `linprog.m` to solve the linear programming problem, their algorithm is restricted to very small images. In [15, 16, 17, 18], Dax investigated regularized least norm problems where the identity matrix is used as the regularization operator (i.e., $R = I$) and developed row-relaxation methods for solving such least norm problems. The row-relaxation method is suitable for solving problems in which H is sparse. On the total variation regularization, Li and Santosa [29] employed both the steepest descent and an affine scaling Newton method, Ito and Kunisch [24] presented an active set strategy based on the augmented Lagrangian formulation, and Hintermüller and Kunisch [23] solved it as a bilaterally constrained optimization problem. Also Combettes [14] used an adaptive level set method for nondifferentiable constrained image recovery.

Usually, images satisfy

$$\mathbf{f} \geq 0,$$

that is, pixels have nonnegative values. However, this constraint is often omitted in restoration methods, mainly because of the numerical intricacies it entails. Hanke, Nagy, and Vogel [22] considered and studied three nonnegatively constrained restorations schemes: constrained least squares, maximum likelihood, and maximum entropy. Saunders [46] used primal-dual interior point methods for nonnegative least squares image restoration problems.

The main contribution of this work is to develop efficient algorithms for the minimization of mixed $\ell 2$-$\ell 1$ and $\ell 1$-$\ell 1$ norms for image restoration problems. The novelty comes by taking into account the nonnegativity constraint. We first formulate (1.4) as a linear programming problem and (1.5) as a quadratic programming problem. Then we solve these problems by combining an interior point method and a preconditioned conjugate gradient method. In each interior point iteration, a structured linear system is obtained and solved. The form of the coefficient matrix of this structured linear system is given by $D + H^T D' H + R^T D'' R$, where H is a BTTB matrix, and D, D', and D'' are diagonal matrices. To solve such inner systems quickly, we employ the preconditioned conjugate gradient method with factorized sparse inverse preconditioners. In [30], it is shown that factorized sparse inverse preconditioners are better than circulant preconditioners for certain image restoration problems. In this paper, we further demonstrate that factorized sparse inverse preconditioners are quite effective for minimization of mixed $\ell 2$-$\ell 1$ and $\ell 1$-$\ell 1$ norms in image restoration problems.

The rest of this paper is outlined as follows. In section 2, we formulate (1.5) as a linear programming problem and solve it by an interior point method. The major work of each iteration of the interior point method is to solve a large structured

linear system. We propose to solve this system by iterative methods. In this section, we also discuss finding an initial feasible point and keeping the iterates feasible. In section 3, we formulate (1.4) as a quadratic programming problem and also solve it by an interior point method. In section 4, we discuss how to precondition the structured linear systems encountered in sections 2 and 3. Experimental results are presented in section 5. Conclusions are made in section 6.

2. The LAD solution.

2.1. Formulating the problem as a linear programming problem. Problem (1.5), with the nonnegativity constraint imposed, can be formulated as a linear programming problem as follows. If the nonnegativity constraint is not desired, only slight modifications of this procedure are needed.

Let $\mathbf{u} = H\mathbf{f} - \mathbf{g}$; let $\mathbf{v} = \alpha R\mathbf{f}$. We split $\mathbf{u}$ and $\mathbf{v}$ into their nonnegative and nonpositive parts. That is, $\mathbf{u} = \mathbf{u}^+ - \mathbf{u}^-$ and $\mathbf{v} = \mathbf{v}^+ - \mathbf{v}^-$, where $\mathbf{u}^+ = \max(\mathbf{u}, 0)$, $\mathbf{u}^- = \max(-\mathbf{u}, 0)$, $\mathbf{v}^+ = \max(\mathbf{v}, 0)$, and $\mathbf{v}^- = \max(-\mathbf{v}, 0)$. The problem can now be written as

$$\begin{aligned} \min_{\mathbf{f},\mathbf{u}^+,\mathbf{u}^-,\mathbf{v}^+,\mathbf{v}^-} \quad & \mathbf{1}^T\mathbf{u}^+ + \mathbf{1}^T\mathbf{u}^- + \mathbf{1}^T\mathbf{v}^+ + \mathbf{1}^T\mathbf{v}^- \\ \text{subject to} \quad & H\mathbf{f} - \mathbf{g} = \mathbf{u}^+ - \mathbf{u}^-, \\ & \alpha R\mathbf{f} = \mathbf{v}^+ - \mathbf{v}^-, \\ & \mathbf{u}^+,\ \mathbf{u}^-,\ \mathbf{v}^+,\ \mathbf{v}^-,\ \mathbf{f} \geq 0. \end{aligned} \tag{2.1}$$

We use $\mathbf{1}$ to denote the vector of all ones of appropriate size. This notation is used throughout this paper.

Clearly (2.1) can be written as a linear programming problem in the standard form:

$$\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x} \quad \text{subject to} \quad A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq 0, \tag{2.2}$$

where A, $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{x}$ are defined by

$$A = \begin{bmatrix} H & -I & I & 0 & 0 \\ \alpha R & 0 & 0 & -I & I \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix},$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{f} \\ \mathbf{u}^+ \\ \mathbf{u}^- \\ \mathbf{v}^+ \\ \mathbf{v}^- \end{bmatrix}, \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} 0 \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}.$$

The Lagrangian function for (2.2) is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) = \mathbf{c}^T\mathbf{x} - \boldsymbol{\lambda}^T(A\mathbf{x} - \mathbf{b}) - \mathbf{s}^T\mathbf{x}. \tag{2.3}$$

Here $\boldsymbol{\lambda}$ and $\mathbf{s}$ are the Lagrange multiplier vectors for the constraints $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq 0$, respectively. For clarity, we partition $\boldsymbol{\lambda}$ as

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_u \\ \boldsymbol{\lambda}_v \end{bmatrix}, \tag{2.4}$$

where $\boldsymbol{\lambda}_u$ is the Lagrange multiplier vector for the constraint $H\mathbf{f} - \mathbf{u}^+ + \mathbf{u}^- = \mathbf{g}$, and $\boldsymbol{\lambda}_v$ is for $\alpha R - \mathbf{v}^+ + \mathbf{v}^- = 0$.

2.2. Interior point methods. Primal-dual interior point methods have become a popular choice for solving large linear programming problems. We briefly outline our adapted interior point method below. A detailed description of interior point methods for linear programming can be found in [49, Chapter 14] or [40, Chapter 1].

The optimality condition for (2.2) is as follows:

$$F(\mathbf{x},\boldsymbol{\lambda},\mathbf{s}) = \begin{bmatrix} A^T\boldsymbol{\lambda} + \mathbf{s} - \mathbf{c} \\ A\mathbf{x} - \mathbf{b} \\ XS\mathbf{1} \end{bmatrix} = 0, \quad \mathbf{x} \geq 0, \quad \mathbf{s} \geq 0, \tag{2.5}$$

where

$$X \;=\; \mathrm{diag}(\mathbf{x}) \quad \text{and} \quad S \;=\; \mathrm{diag}(\mathbf{s}). \tag{2.6}$$

Primal-dual interior point methods have their origin in Newton's method for the system of nonlinear equations (2.5). Newton's method starts with some initial guess for the solution and calculates a search direction at each iteration by solving a linearized model of the original system. A detailed description of Newton's method for nonlinear systems can be found in [40, Chapter 11]. In the primal-dual interior point algorithm, the basic Newton step is modified such that the search directions are aimed at points on the *central path* $(\mathbf{x}_\tau, \boldsymbol{\lambda}_\tau, \mathbf{s}_\tau)$, which is defined as

$$F(\mathbf{x}_\tau,\boldsymbol{\lambda}_\tau,\mathbf{s}_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau\mathbf{1} \end{bmatrix}, \quad \mathbf{x}_\tau > 0, \quad \mathbf{s}_\tau > 0. \tag{2.7}$$

Very often τ is written as $\sigma\mu$, where $\sigma \in [0,1]$ is a *centering parameter*, and μ is the *duality measure* defined by

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i s_i = \frac{\mathbf{x}^T\mathbf{s}}{n}. \tag{2.8}$$

The step length at each iteration is chosen such that the new iterate is strictly positive, that is, $\mathbf{x} > 0$ and $\mathbf{s} > 0$.

The Newton search direction, $(\triangle\mathbf{x}, \triangle\boldsymbol{\lambda}, \triangle\mathbf{s})$, is computed by solving the following linear system:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \triangle\mathbf{x} \\ \triangle\boldsymbol{\lambda} \\ \triangle\mathbf{s} \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_c \\ -\mathbf{r}_b \\ -\mathbf{r}_a \end{bmatrix}, \tag{2.9}$$

where

$$\mathbf{r}_c \;=\; A^T\boldsymbol{\lambda} + \mathbf{s} - \mathbf{c}, \quad \mathbf{r}_b \;=\; A\mathbf{x} - \mathbf{b}, \quad \mathbf{r}_a \;=\; XS\mathbf{1} - \sigma\mu\mathbf{1}.$$

By eliminating $\triangle\mathbf{s}$ in (2.9), we obtain

$$\begin{bmatrix} -X^{-1}S & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \triangle\mathbf{x} \\ \triangle\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\hat{\mathbf{r}}_c \\ -\mathbf{r}_b \end{bmatrix}, \tag{2.10}$$

where $\hat{\mathbf{r}}_c = \mathbf{r}_c - X^{-1}\mathbf{r}_a$. Let $D = S^{-1/2}X^{1/2}$; then (2.10) can be written as

$$\begin{bmatrix} -D^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \triangle\mathbf{x} \\ \triangle\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\hat{\mathbf{r}}_c \\ -\mathbf{r}_b \end{bmatrix}. \tag{2.11}$$

Note that since the components of $\mathbf{x}$ and $\mathbf{s}$ are strictly positive, all the diagonal elements of D are well defined.

From here the standard approach is to reduce (2.11) to the following form and solve it for step $\triangle\boldsymbol{\lambda}$:

$$AD^2A^T\triangle\boldsymbol{\lambda} = -AD^2\hat{\mathbf{r}}_c - \mathbf{r}_b. \tag{2.12}$$

The application of the preconditioned MINRES [41] or the preconditioned GMRES [45] to (2.10) may be considered. For example, the preconditioner of Benzi and Golub [7], which is briefly described in section 4, could be used to solve (2.10).

However, after comparing the performance of the different implementations, we found that it is advantageous to proceed as follows. System (2.11) can be written as

$$\begin{bmatrix} -D_1^{-2} & 0 & 0 & 0 & 0 & H^T & \alpha R^T \\ 0 & -D_2^{-2} & 0 & 0 & 0 & -I & 0 \\ 0 & 0 & -D_3^{-2} & 0 & 0 & I & 0 \\ 0 & 0 & 0 & -D_4^{-2} & 0 & 0 & -I \\ 0 & 0 & 0 & 0 & -D_5^{-2} & 0 & I \\ H & -I & I & 0 & 0 & 0 & 0 \\ \alpha R & 0 & 0 & -I & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \triangle\mathbf{f} \\ \triangle\mathbf{u}^+ \\ \triangle\mathbf{u}^- \\ \triangle\mathbf{v}^+ \\ \triangle\mathbf{v}^- \\ \triangle\boldsymbol{\lambda}_u \\ \triangle\boldsymbol{\lambda}_v \end{bmatrix} = \begin{bmatrix} -\hat{\mathbf{r}}_{c1} \\ -\hat{\mathbf{r}}_{c2} \\ -\hat{\mathbf{r}}_{c3} \\ -\hat{\mathbf{r}}_{c4} \\ -\hat{\mathbf{r}}_{c5} \\ -\mathbf{r}_{b1} \\ -\mathbf{r}_{b2} \end{bmatrix}.$$

Here D_i (for $i = 1, 2, 3, 4, 5$) are submatrices of D of the appropriate size, and $\hat{\mathbf{r}}_{ci}$ (for $i = 1, 2, 3, 4, 5$) and $\mathbf{r}_{bi}$ (for $i = 1, 2$) are subvectors of the appropriate size of $\hat{\mathbf{r}}_c$ and $\mathbf{r}_b$, respectively.

By eliminating $\triangle\mathbf{u}^+$, $\triangle\mathbf{u}^-$, $\triangle\mathbf{v}^+$, and $\triangle\mathbf{v}^-$, we obtain

$$\begin{bmatrix} -D_1^{-2} & H^T & \alpha R^T \\ H & D_2^2 + D_3^2 & 0 \\ \alpha R & 0 & D_4^2 + D_5^2 \end{bmatrix} \begin{bmatrix} \triangle\mathbf{f} \\ \triangle\boldsymbol{\lambda}_u \\ \triangle\boldsymbol{\lambda}_v \end{bmatrix} = \begin{bmatrix} -\hat{\mathbf{r}}_{c1} \\ -\hat{\mathbf{r}}_{b1} \\ -\hat{\mathbf{r}}_{b2} \end{bmatrix}, \tag{2.13}$$

where

$$\hat{\mathbf{r}}_{b1} = \mathbf{r}_{b1} - D_2^2\hat{\mathbf{r}}_{c2} + D_3^2\hat{\mathbf{r}}_{c3} \quad \text{and} \quad \hat{\mathbf{r}}_{b2} = \mathbf{r}_{b2} - D_4^2\hat{\mathbf{r}}_{c4} + D_5^2\hat{\mathbf{r}}_{c5}.$$

By eliminating $\triangle\boldsymbol{\lambda}_u$ and $\triangle\boldsymbol{\lambda}_v$ from (2.13), we obtain

$$\left[D_1^{-2} + H^T(D_2^2 + D_3^2)^{-1}H + \alpha^2 R^T(D_4^2 + D_5^2)^{-1}R\right]\triangle\mathbf{f} = \tilde{\mathbf{r}}_{c1}, \tag{2.14}$$

where

$$\tilde{\mathbf{r}}_{c1} = \hat{\mathbf{r}}_{c1} - H^T(D_2^2 + D_3^2)^{-1}\hat{\mathbf{r}}_{b1} - \alpha R^T(D_4^2 + D_5^2)^{-1}\hat{\mathbf{r}}_{b2}.$$

System (2.14) can be viewed as the normal equations for a linear least squares problem with the following coefficient matrix:

$$\begin{bmatrix} D_1^{-1} \\ (D_2^2 + D_3^2)^{-1/2}H \\ \alpha(D_4^2 + D_5^2)^{-1/2}R \end{bmatrix} = \begin{bmatrix} D_1^{-1} & & \\ & (D_2^2 + D_3^2)^{-1/2} & \\ & & (D_4^2 + D_5^2)^{-1/2} \end{bmatrix} \begin{bmatrix} I \\ H \\ \alpha R \end{bmatrix}. \tag{2.15}$$

Clearly the second factor in the right-hand side of (2.15) has full column rank and is well-conditioned. Since the matrices D_i are all diagonal matrices with positive diagonal elements, the first factor in the right-hand side of (2.15) has full column

rank. It follows that the coefficient matrix in (2.14) is symmetric positive definite and can be solved by the conjugate gradient method.

In general, when the iterates are close to the solution, if the primal variable is close to zero, then the corresponding dual variable will be far away from zero. This means the corresponding element in D will be very small. On the other hand, if the primal variable is far away from zero, then the corresponding dual variable will be very close to zero, leading to a large entry in D. Therefore, when the iterates are close to the solution, the matrix D will have both huge and tiny elements, causing ill-conditioning. In section 4, we discuss how to precondition system (2.14) efficiently.

Once we have solved (2.14) for $\triangle\mathbf{f}$, the other unknowns in (2.9) can be recovered by the following equations:

$$\begin{cases} \triangle\boldsymbol{\lambda}_u = (D_2^2 + D_3^2)^{-1}(-\hat{\mathbf{r}}_{b1} - H\triangle\mathbf{f}), \\ \triangle\boldsymbol{\lambda}_v = (D_4^2 + D_5^2)^{-1}(-\hat{\mathbf{r}}_{b2} - \alpha R\triangle\mathbf{f}), \\ \triangle\mathbf{u}^+ = D_2^2(\hat{\mathbf{r}}_{c2} - \triangle\boldsymbol{\lambda}_u), \\ \triangle\mathbf{u}^- = D_3^2(\hat{\mathbf{r}}_{c3} + \triangle\boldsymbol{\lambda}_u), \\ \triangle\mathbf{v}^+ = D_4^2(\hat{\mathbf{r}}_{c4} - \triangle\boldsymbol{\lambda}_v), \\ \triangle\mathbf{v}^- = D_5^2(\hat{\mathbf{r}}_{c5} + \triangle\boldsymbol{\lambda}_v), \\ \triangle\mathbf{s} = -\mathbf{r}_c - A^T\triangle\boldsymbol{\lambda}. \end{cases} \tag{2.16}$$

2.3. Finding an initial feasible point and staying feasible. Define the *strictly feasible set* $\mathcal{F}^0$ as

$$\mathcal{F}^0 = \{(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) | A\mathbf{x} = \mathbf{b}, A^T\boldsymbol{\lambda} + \mathbf{s} = \mathbf{c}, \mathbf{x} > 0, \mathbf{s} > 0\}. \tag{2.17}$$

Finding an initial strictly feasible point is not straightforward in many linear programming problems. For our problem, an initial strictly feasible point can be found as follows. Let ϵ be some positive value. Let $\mathbf{f}_{ls}$ be any estimate of the sought-after solution, for example, the least squares solution. We choose $\mathbf{f}^{(0)}$ to be

$$\mathbf{f}^{(0)}[i] = \max\{\mathbf{f}_{ls}[i], \epsilon\}.$$

We let $\mathbf{u}^{+(0)}$ be the sum of ϵ and the nonnegative part of $H\mathbf{f}^{(0)}$ and we let $\mathbf{u}^{-(0)}$ be the sum of ϵ and the nonpositive part of $H\mathbf{f}^{(0)}$. Then the vector

$$\mathbf{x}^{(0)} = \begin{bmatrix} \mathbf{f}^{(0)} \\ \mathbf{u}^{+(0)} \\ \mathbf{u}^{-(0)} \\ \mathbf{v}^{+(0)} \\ \mathbf{v}^{-(0)} \end{bmatrix}$$

satisfies $A\mathbf{x}^{(0)} = \mathbf{b}$. Furthermore, $\mathbf{x}^{(0)} \geq \epsilon$. There are many ways that we can find $\boldsymbol{\lambda}^{(0)}$ and $\mathbf{s}^{(0)}$ such that $\mathbf{s}^{(0)} > 0$ and $A^T\boldsymbol{\lambda}^{(0)} + \mathbf{s}^{(0)} - \mathbf{c} = 0$. Since our focus is on the problem where H corresponds to a shift invariant blurring operator, we can let $\boldsymbol{\lambda}_u^{(0)}$ be a constant vector of -0.5 and let $\boldsymbol{\lambda}_v^{(0)}$ be the zero vector. If Neumann boundary conditions are used, and the blurring function is normalized such that it integrates to 1, then $H^T\boldsymbol{\lambda}_u^{(0)} = 0.5 \cdot \mathbf{1}$. Therefore, if we let $\mathbf{s}^{(0)} = \mathbf{c} - A^T\boldsymbol{\lambda}^{(0)}$, then $\mathbf{s}^{(0)} \geq 0.5 \cdot \mathbf{1}$, and $(\mathbf{x}^{(0)}, \boldsymbol{\lambda}^{(0)}, \mathbf{s}^{(0)})$ is strictly feasible. If zero boundary conditions are used, this procedure would still give an initial strictly feasible point under weak assumptions on the blurring function.

If the current iterate is in the strictly feasible set, and (2.9) is solved exactly, and if a step length is chosen such that the new iterate satisfies $\mathbf{x} > 0$ and $\mathbf{s} > 0$, then the new iterate is strictly feasible.

However, when we use the conjugate gradient method to solve (2.14), only an approximate solution is given. Therefore, even if the current iterate is strictly feasible, the new iterate may not satisfy the conditions

$$A\mathbf{x} = \mathbf{b} \quad \text{and} \quad A^T\boldsymbol{\lambda} + \mathbf{s} - \mathbf{c} = 0$$

to very high accuracy. Thus, if no precaution is taken, the subsequent iterates could move further from the feasible set due to error propagation.

Fortunately, it is easy to show that if (2.16) is used to recover the other unknowns after (2.14) is solved, the conditions $A\mathbf{x} = \mathbf{b}$ and $A^T\boldsymbol{\lambda} + \mathbf{s} - \mathbf{c} = 0$ will be satisfied to machine precision even if (2.14) is solved inexactly. Therefore, if a proper step length is chosen, the next iterate will be strictly feasible.

2.4. Summary of the adapted interior point algorithm. There are many variants of the interior point method. Our algorithm, summarized below, is based on the simplest framework, that is, framework PD in [49, p. 8].

ALGORITHM. INTERIOR POINT FOR THE LAD SOLUTION.

1. Initialize $(\mathbf{x}^{(0)}, \boldsymbol{\lambda}^{(0)}, \mathbf{s}^{(0)}) \in \mathcal{F}^o$.
2. for $k = 0, 1, 2 \ldots$
   - choose $\sigma \in [0, 1]$.
   - solve (2.14) for $\triangle\mathbf{f}$.
   - Use (2.16) to recover $\triangle\mathbf{u}^+$, $\triangle\mathbf{u}^-$, $\triangle\mathbf{v}^+$, $\triangle\mathbf{v}^-$, $\triangle\boldsymbol{\lambda}$, $\triangle\mathbf{s}$.
   - choose β^{pri} and β^{dual}.
   - $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \beta^{pri}\triangle\mathbf{x}$
   - $\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \beta^{dual}\triangle\boldsymbol{\lambda}$
   - $\mathbf{s}^{(k+1)} = \mathbf{s}^{(k)} + \beta^{dual}\triangle\mathbf{s}$

   end

Since we use iterative methods to solve the linear systems, no predictor-corrector strategy is involved.

A word needs to be said about the choices of centering parameter σ and step lengths β^{pri} and β^{dual}. It is a common practice to choose the primal and dual step lengths β^{pri} and β^{dual} as follows:

$$\beta^{pri} = \min\left(1, \eta\beta^{pri}_{max}\right), \quad \beta^{dual} = \min\left(1, \eta\beta^{dual}_{max}\right).$$

Here β^{pri}_{max} and β^{dual}_{max} are the maximum steps that can be taken before violating the nonnegativity condition, and $\eta \in [0.9,\ 1.0)$. We simply choose $\eta = 0.95$. Our experiments led us to choose the centering parameter σ to be 0.1 at the first iteration, and

$$\sigma = \max(0.01, (1 - \min(\beta^{pri}, \beta^{dual}))^3)$$

in other iterations. This prevents the iterates from moving too far from the central path and allows significant reduction in duality at each iteration.

3. The LMN solution. Problem (1.4), with the nonnegativity constraint imposed, can be restated as a quadratic programming problem as follows. We let

$\mathbf{v} = \alpha R\mathbf{f}$, and split $\mathbf{v}$ into its nonnegative and nonpositive parts $\mathbf{v}^+$ and $\mathbf{v}^-$. Then (1.4) can be written as

$$\begin{aligned} \min_{\mathbf{f},\mathbf{v}^+,\mathbf{v}^-} \; & \mathbf{1}^T\mathbf{v}^+ + \mathbf{1}^T\mathbf{v}^- + \|H\mathbf{f} - \mathbf{g}\|_2^2 \\ \text{subject to} \; & \alpha R\mathbf{f} = \mathbf{v}^+ - \mathbf{v}^-, \\ & \mathbf{v}^+, \mathbf{v}^-, \mathbf{f} \geq 0. \end{aligned} \tag{3.1}$$

Problem (3.1) can be written as

$$\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^T G\mathbf{x} + \mathbf{c}^T\mathbf{x} \quad \text{subject to} \quad A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq 0, \tag{3.2}$$

where G, A, $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{x}$ are defined by

$$G = \begin{bmatrix} 2H^TH & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} \alpha R & -I & I \end{bmatrix},$$

$$\mathbf{b} = \mathbf{0}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{f} \\ \mathbf{v}^+ \\ \mathbf{v}^- \end{bmatrix}, \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} -2H^T\mathbf{g} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}.$$

We remark that the objective function is convex since G is symmetric positive semidefinite. Problem (3.2) can also be solved by an interior point method. The Lagrangian function for (3.2) is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) = \frac{1}{2}\mathbf{x}^T G\mathbf{x} + \mathbf{c}^T\mathbf{x} - \boldsymbol{\lambda}^T(A\mathbf{x} - \mathbf{b}) - \mathbf{s}^T\mathbf{x}, \tag{3.3}$$

where $\boldsymbol{\lambda}$ and $\mathbf{s}$ are the generalized Lagrange multiplier vectors for the constraints $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq 0$, respectively.

The optimality condition for (3.2) is

$$F(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) = \begin{bmatrix} G\mathbf{x} + \mathbf{c} - A^T\boldsymbol{\lambda} - \mathbf{s} \\ A\mathbf{x} - \mathbf{b} \\ XS\mathbf{1} \end{bmatrix} = 0, \quad \mathbf{x} \geq 0, \quad \mathbf{s} \geq 0. \tag{3.4}$$

The central path, $(\mathbf{x}_{\sigma\mu}, \boldsymbol{\lambda}_{\sigma\mu}, \mathbf{s}_{\sigma\mu})$, is defined similarly as in the previous section:

$$F(\mathbf{x}_{\sigma\mu}, \boldsymbol{\lambda}_{\sigma\mu}, \mathbf{s}_{\sigma\mu}) = \begin{bmatrix} 0 \\ 0 \\ \sigma\mu\mathbf{1} \end{bmatrix}. \tag{3.5}$$

Recall that $\sigma \in (0, 1)$ and μ is defined as in (2.8). The Newton step, $(\triangle\mathbf{x}, \triangle\boldsymbol{\lambda}, \triangle\mathbf{s})$, is computed by solving the following linear system:

$$\begin{bmatrix} G & -A^T & -I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \triangle\mathbf{x} \\ \triangle\boldsymbol{\lambda} \\ \triangle\mathbf{s} \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_c \\ -\mathbf{r}_b \\ -\mathbf{r}_a \end{bmatrix}, \tag{3.6}$$

where

$$\mathbf{r}_c = G\mathbf{x} + \mathbf{c} - A^T\boldsymbol{\lambda} - \mathbf{s}, \quad \mathbf{r}_b = A\mathbf{x} - \mathbf{b}, \quad \mathbf{r}_a = XS\mathbf{1} - \sigma\mu\mathbf{1}.$$

By eliminating $\triangle \mathbf{s}$, we obtain

$$\begin{bmatrix} G + X^{-1}S & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \triangle \mathbf{x} \\ \triangle \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\hat{\mathbf{r}}_c \\ -\mathbf{r}_b \end{bmatrix}, \tag{3.7}$$

where $\hat{\mathbf{r}}_c = \mathbf{r}_c + X^{-1}\mathbf{r}_a$. This system can be made symmetric by a change of variable:

$$\begin{bmatrix} G + X^{-1}S & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \triangle \mathbf{x} \\ -\triangle \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\hat{\mathbf{r}}_c \\ -\mathbf{r}_b \end{bmatrix}. \tag{3.8}$$

Let $D = S^{-1/2}X^{1/2}$. Note that the elements of D are well defined. Partition D as

$$D = \ \mathrm{diag}(D_1, D_2, D_3),$$

where D_1, D_2, and D_3 are matrices of appropriate size such that system (3.8) can be put into the following form:

$$\begin{bmatrix} 2H^TH + D_1^{-2} & 0 & 0 & \alpha R^T \\ 0 & D_2^{-2} & 0 & -I \\ 0 & 0 & D_3^{-2} & I \\ \alpha R & -I & I & 0 \end{bmatrix} \begin{bmatrix} \triangle \mathbf{f} \\ \triangle \mathbf{v}^+ \\ \triangle \mathbf{v}^- \\ -\triangle \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\hat{\mathbf{r}}_{c1} \\ -\hat{\mathbf{r}}_{c2} \\ -\hat{\mathbf{r}}_{c3} \\ -\mathbf{r}_b \end{bmatrix}. \tag{3.9}$$

Here $\hat{\mathbf{r}}_{ci}$ (for $i = 1,\ 2,\ 3$) are subvectors of the appropriate size of $\hat{\mathbf{r}}_c$. By eliminating $\triangle \mathbf{v}^+$ and $\triangle \mathbf{v}^-$, we get

$$\begin{bmatrix} 2H^TH + D_1^{-2} & \alpha R^T \\ \alpha R & -D_2^2 - D_3^2 \end{bmatrix} \begin{bmatrix} \triangle \mathbf{f} \\ -\triangle \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\hat{\mathbf{r}}_{c1} \\ -\hat{\mathbf{r}}_b \end{bmatrix}, \tag{3.10}$$

where

$$\hat{\mathbf{r}}_b = \mathbf{r}_b + D_2^2\hat{\mathbf{r}}_{c2} - D_3^2\hat{\mathbf{r}}_{c3}.$$

Eliminating $\triangle \boldsymbol{\lambda}$, we get

$$\left[2H^TH + D_1^{-2} + \alpha^2 R^T(D_2^2 + D_3^2)^{-1}R\right] \triangle \mathbf{f} = -\tilde{\mathbf{r}}_{c1}, \tag{3.11}$$

where

$$\tilde{\mathbf{r}}_{c1} = \hat{\mathbf{r}}_{c1} + \alpha R^T(D_2^2 + D_3^2)^{-1}\hat{\mathbf{r}}_b.$$

Similar to system (2.14), system (3.11) can be viewed as the normal equations of a linear least squares problem, whose coefficient matrix has full column rank. Therefore, the coefficient matrix in (3.11) is symmetric positive definite, and system (3.11) can be solved by the preconditioned conjugate gradient method.

Once (3.11) has been solved, the other unknowns can be calculated by the following equations:

$$\begin{cases} \triangle \boldsymbol{\lambda} = (D_2^2 + D_3^2)^{-1}(-\hat{\mathbf{r}}_b - \alpha R \triangle \mathbf{f}), \\ \triangle \mathbf{v}^+ = D_2^2(-\hat{\mathbf{r}}_{c2} - \triangle \boldsymbol{\lambda}), \\ \triangle \mathbf{v}^- = D_3^2(-\hat{\mathbf{r}}_{c3} + \triangle \boldsymbol{\lambda}), \\ \triangle \mathbf{s} = G \triangle \mathbf{x} - A^T \triangle \boldsymbol{\lambda} + \mathbf{r}_c. \end{cases} \tag{3.12}$$

An initial feasible point can be found in a similar manner as was done for the LAD solution. Moreover, if (3.12) is used to recover the rest of the unknowns after (3.11) is solved approximately, and a proper step length is chosen, then the iterates will stay feasible.

The interior point algorithm for the LMN solution is quite similar to that for the LAD solution. One difference that is worth pointing out is that β^{pri} and β^{dual} are allowed to take different values in the algorithm for the LAD solution but have to be the same in that for the LMN solution.

4. Preconditioning the structured linear systems. As mentioned earlier, linear systems (2.14) and (3.11) become ill-conditioned as the iterates get close to the solution. Thus preconditioners are needed to accelerate the convergence of the conjugate gradient iterations.

For the nonnegative least squares image restoration problem, Saunders [46] used preconditioned LSQR to solve the linear system at each interior point iteration. He used a diagonal preconditioner. However, the effectiveness of diagonal preconditioners is limited in that preconditioned LSQR needed more than 1000 iterations to converge to a desired level of accuracy at the final stages of the interior point algorithm. Our experimental results indicate that the conditions of systems (2.14) and (3.11) tend to be worse than that of the systems in the nonnegative least squares problem. This should be no surprise since our problems are transformed from nonsmooth problems.

4.1. Saddle point preconditioners. For linear systems (2.10), Benzi and Golub [7] proposed using the preconditioned GMRES to solve the equivalent system

$$\begin{bmatrix} D^{-2} & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \triangle \mathbf{x} \\ \triangle \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{r}}_c \\ -\mathbf{r}_b \end{bmatrix}. \tag{4.1}$$

Their preconditioning strategy is based on the symmetric/skew-symmetric splitting of the coefficient matrix, that is, they propose using the following matrix as the preconditioner:

$$\mathcal{M}_\gamma = \begin{bmatrix} D^{-2} + \gamma I & 0 \\ 0 & \gamma I \end{bmatrix} \begin{bmatrix} \gamma I & -A^T \\ A & \gamma I \end{bmatrix}, \tag{4.2}$$

where γ is some positive constant. Linear systems in the form of $\mathcal{M}_\gamma \mathbf{z} = \mathbf{r}$ can be solved by first solving

$$\begin{bmatrix} D^{-2} + \gamma I & 0 \\ 0 & \gamma I \end{bmatrix} \tilde{\mathbf{z}} = \mathbf{r}, \tag{4.3}$$

followed by

$$\begin{bmatrix} \gamma I & -A^T \\ A & \gamma I \end{bmatrix} \mathbf{z} = \tilde{\mathbf{z}}. \tag{4.4}$$

Solving (4.3) is a simple dotwise division. Since our focus is on the shift invariant blur, system (4.4) can be solved by the discrete cosine transform if the blur is symmetric and Neumann boundary conditions are applied; see [36], for instance. This preconditioning technique can also be applied to (2.13), which yields better results in practice.

4.2. Factorized sparse inverse preconditioner. Our experimental results indicate that the most efficient way to compute the Newton step is to apply the factorized sparse inverse preconditioner (FSIP) [27, 8, 47] to systems (2.14) and (3.11). Let B be a symmetric positive definite matrix, and let its Cholesky factorization be $B = CC^T$. The idea of FSIP is to find the lower triangular matrix L with the sparsity pattern $\mathcal{S}$ such that

$$\|I - CL\|_F$$

is minimized, where $\|\cdot\|_F$ denotes the Frobenius norm. Kolotilina and Yeremin [27] showed that L can be obtained by the following algorithm.

Algorithm. Construction of FSIP.
1. Compute $\hat{L}$ with sparsity pattern $\mathcal{S}$ such that $\left[\hat{L}B\right]_{ij} = \delta_{ij}, \quad (i,j) \in \mathcal{S}$.
2. Let $\hat{D} = (\mathrm{diag}(\hat{L}))^{-1}$ and $L = \hat{D}^{1/2}\hat{L}$.

According to this algorithm, n small linear systems need to be solved, where n is the number of rows in the matrix B. These systems can be solved in parallel. Thus the above algorithm is well suited for modern parallel computing.

A special type of FSIP, the factorized banded inverse preconditioner (FBIP), was proposed by Lin, Ng, and Ching [30] for Toeplitz related matrices. We say that $B \in \mathbb{R}^{m\times n}$ has *lower bandwidth* p if $b_{ij} = 0$ whenever $i > j + p$. The *upper bandwidth* of a matrix is defined similarly. Note that by this definition, a tridiagonal matrix has upper and lower bandwidth of 1.

Lin, Ng, and Ching [30] proved the following theorem.

Theorem 4.1. *Let T be a Hermitian Toeplitz matrix. Denote the kth diagonal of T by t_k. Assume the diagonals of T satisfy*

$$|t_k| \le ce^{-\gamma|k|} \tag{4.5}$$

for some $c > 0$ and $\gamma > 0$, or

$$|t_k| \le c(|k| + 1)^{-s} \tag{4.6}$$

for some $c > 0$ and $s > 3/2$. Then for any given $\epsilon > 0$, there exists $p' > 0$ such that for all $p > p'$,

$$\|L_p - C^{-1}\| \le \epsilon,$$

where L_p is the FBIP of T with the lower bandwidth p, and C is the Cholesky factor of T.

This theorem indicates that if a Toeplitz matrix T has certain off-diagonal decay property, then the FBIP will be a good approximation to T^{-1}. Note that if a Toeplitz matrix is banded, then both (4.5) and (4.6) are satisfied trivially.

Lin, Ng, and Ching [30] also considered Toeplitz-related systems of the form $I + T^T DT$, where D is a positive diagonal matrix, and proved the following corollary.

Corollary 4.2. *Let T be a Toeplitz matrix with its diagonals satisfying* (4.5) *or* (4.6). *Let $D = \mathrm{diag}(d_i)$ with $0 < d_i \le d$, and*

$$B = I + T^T DT.$$

Then for any given $\epsilon > 0$, there exists $p' > 0$ such that for all $p > p'$,

$$\|L_p - C^{-1}\| \leq \epsilon,$$

where L_p is the FBIP of B with the lower bandwidth p, and C is the Cholesky factor of B.

The matrices we are trying to precondition are the coefficient matrices in (2.14) and (3.11), that is, $B = D_1^{-2} + H^T(D_2^2 + D_3^2)^{-1}H + \alpha^2 R^T(D_4^2 + D_5^2)^{-1}R$ and $B = 2H^T H + D_1^{-2} + \alpha^2 R^T(D_2^2 + D_3^2)^{-1}R$. Although the focus of this research is on two-dimensional images, we have the following lemma for the one-dimensional problem.

LEMMA 4.3. *Let T be a Toeplitz matrix with its diagonals satisfying* (4.5) *or* (4.6). *Let D_1, D_2, and D_3 be diagonal matrices with positive diagonal entries. Let*

$$B = T^T D_1 T + D_2 + R^T D_3 R \tag{4.7}$$

be a well-conditioned matrix, where R is the first order or the second order difference operator. Then for any given $\epsilon > 0$, there exists $p' > 0$ such that for all $p > p'$,

$$\|L_p - C^{-1}\| \leq \epsilon,$$

where L_p is the FBIP of B with lower bandwidth p, and C is the Cholesky factor of B.

Proof. The proof is similar to the proof of Corollary 4 in [30]. ∎

Lemma 4.3 indicates that the linear system in the one-dimensional LMN and LAD problems can be efficiently preconditioned by the FBIP.

It is important to note that in order to construct the FBIP for B, it is not necessary to explicitly calculate all the elements of B when p is small. The following lemma indicates that only certain diagonals of B need to be calculated.

LEMMA 4.4. *Let L_p be the FBIP with lower bandwidth p for a symmetric matrix B. Then L_p can be determined by the first p bands of the lower triangular part of B.*

Proof. For $i = 1, 2, \ldots, p-1$, the nonzero elements of the ith row of L_p is $L(i, 1:i)$. They are determined by $B(1:i, 1:i)$.

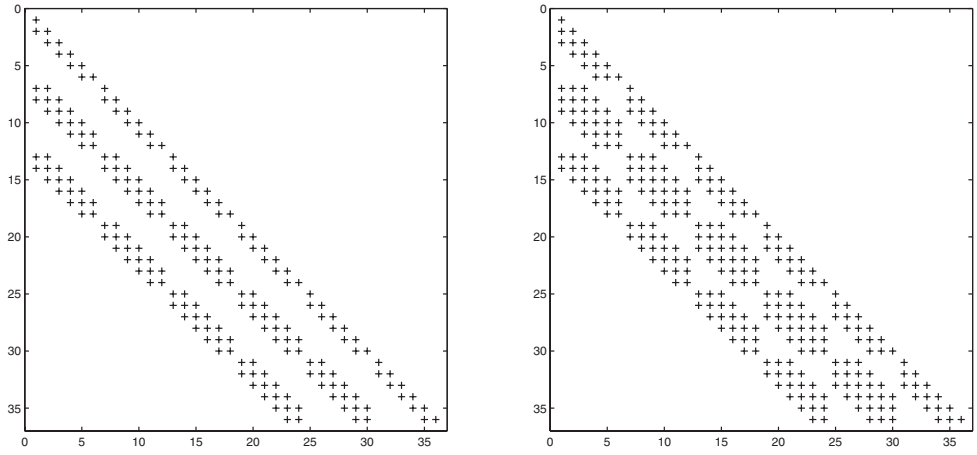
On the other hand, for $i = p, p+1, \ldots, n$, the nonzero elements of the ith row of L_p is $L(i, i+1-p:i)$. They are determined by $B(i+1-p:i, i+1-p:i)$.

Therefore, L_p can be determined by the first upper and lower p bands of B. Since B is symmetric, the result follows. ∎

For two-dimensional problems, we assume that the blurring matrix H has a block-level off-diagonal decay property, and each block also has off-diagonal decay property. This is true if the blurring function decays in spatial domain, or if the support of the blur is small. In this case, we can set the FSIP to be a triangular block banded matrix with each block being a banded matrix. Let p be the block-level lower bandwidth of the FSIP and let q be the lower and the upper bandwidths of each block. Figure 4.1(a) shows the sparsity pattern of an FSIP with $p = 2$ and $q = 1$. Note that if $p = q = 0$, then the FSIP becomes a diagonal preconditioner.

Similar to the one-dimensional case, when constructing the FSIP, it is not necessary to calculate all the elements of the coefficient matrix in (2.14) or (3.11). However, the situation for the two-dimensional case is a bit more complicated.

LEMMA 4.5. *Let L be the FSIP of the coefficient matrix in* (2.14) *or* (3.11), *and let p be the block-level lower bandwidth and q the upper and the lower bandwidths of the blocks of L. Then L can be determined by those elements that are in the lower*



(a) (b)

FIG. 4.1. (a) *FSIP with* $p = 2$ *and* $q = 1$. (b) *Sparsity pattern of the elements of the coefficient matrix that needs to be calculated.*

triangular part of the coefficient matrix and have a block-level lower bandwidth of p *with each block having upper and lower bandwidth of* $2q$.

Lemma 4.5 is actually a bit conservative. Figure 4.1(b) shows the sparsity pattern of the elements of the coefficient matrix that needs to be calculated if the FSIP has a sparsity pattern shown in Figure 4.1(a). Note that the blocks in the last block diagonal have lower bandwidth of q. If the original blurring matrix H is sparse enough such that there is no need to use the FFT [31] for matrix multiplications, the cost of calculating these elements is no more than $(p+1)(2q+1)$ plain conjugate gradient iterations.

Other than explicitly calculating certain diagonals of the coefficient matrix in (2.14) or (3.11), the cost of constructing the FSIP consists of calculating the lower triangular factor L. Since this step solves mn linear systems of size $(p+1)(2q+1)\times(p+1)(2q+1)$, the complexity of this step is $O(p^3q^3mn)$. Because this cost grows cubically with the increase of p and q, the benefit of the decrease in the number of iterations will be offset by the cost of constructing the preconditioner with the increase of p and q. Therefore, it is important to choose suitable values of p and q in practical applications.

Usually the linear system to be solved is well-conditioned at the first few interior point iterations. Due to the initial cost of constructing the FSIP, it is too expensive to choose a big p or q for early iterates. However, as the iterates get close to the solution, the ill-conditioning of the linear system makes choosing bigger p and q necessary. Therefore, the best strategy is to choose small p and q (say $p = q = 0$) at the beginning, and increase p and q gradually. For details, see experimental results in the next section.

5. Experimental results. Most of our experiments are based on the cameraman image shown in Figure 5.1. The blurring function is chosen to be a two-dimensional Gaussian,

$$h(i,j) = e^{-2(i/3)^2-2(j/3)^2},$$

truncated such that the function has a support of 7×7.

We generate two observed images. For the first one, a Gaussian white noise with standard deviation 0.05 is added. For the second one, we randomly pick 50% of the pixels to be contaminated by noise; the type and standard deviation of the noise is the same as in the first one. The observed images are shown in Figure 5.2. For the

FIG. 5.1. *The original image.*

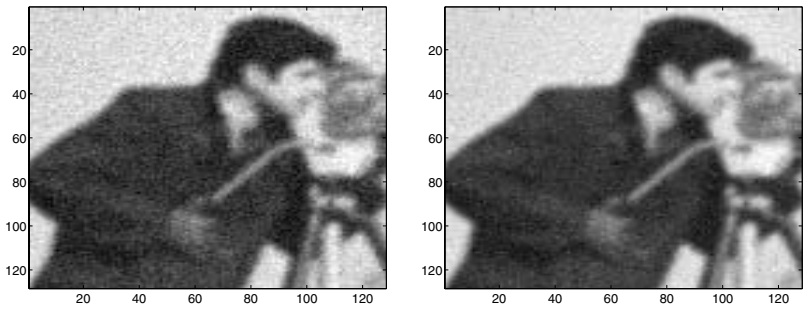


FIG. 5.2. *Observed images:* (a) *all pixels are contaminated by noise;* (b) *only* 50% *of the pixels are contaminated by noise.*

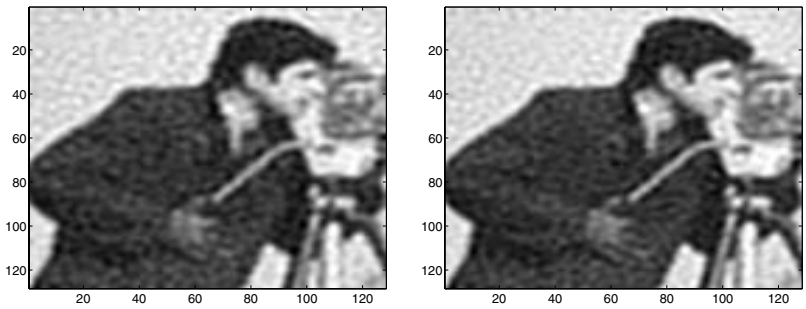


FIG. 5.3. (a) *The least squares solutions for the first observed image, PSNR* = 20.46 *db.* (b) *The least squares solution for the second observed image, PSNR* = 20.87 *db.*

first observed image, since the noise is normal, we consider only the LMN solution. For the second one, we consider both the LMN solution and the LAD solution.

Neumann boundary conditions are used to build the blurring matrix H. Figure 5.3 shows the least squares restoration of the two observed images, respectively. The optimal regularization parameter (the one that yields the highest peak signal-to-noise ratio (PSNR)) is used.

The interior point algorithms are implemented in MATLAB, except the construction of the FSIP, which is written as a MATLAB callable C function. Recall that in order to construct the FSIP, mn small linear systems need to be solved. Doing this

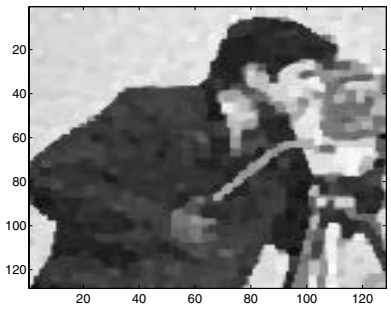


FIG. 5.4. *The LMN solution for the first observed image, PSNR* = 20.78 *db.*

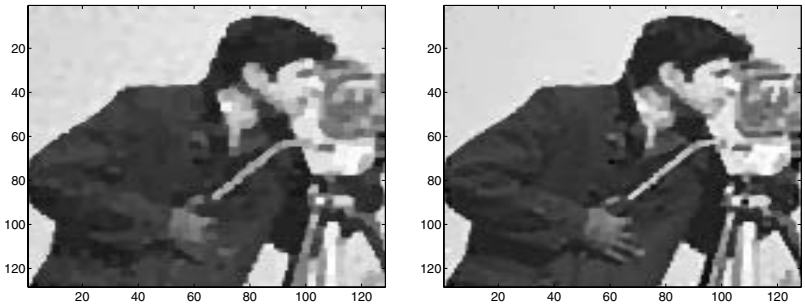


FIG. 5.5. (a) *The LMN solution for the second observed image, PSNR* = 21.30 *db.* (b) *The LAD solution for the second observed image, PSNR* = 22.82 *db.*

in MATLAB would be slow because MATLAB loops are inefficient.

The LMN solution for the first observed image is shown in Figure 5.4. The PSNR of this image indicates it is superior to the least squares solution. It is interesting to note that the LMN solution is quite different from the least squares solution. The least squares solution tends to smooth out the edges but undersmooths flat areas. The LMN solution tends to sharpen the edges and makes flat areas even flatter.

The LAD and LMN solutions for the second observed image is shown in Figure 5.5. As can be seen, both solutions have a higher PSNR than the least squares solution. Furthermore, the LAD solution is superior to the LMN solution.

Similar to the total variation deblurring [43], using the ℓ_1 norm for the regularization term is very effective in recovering "blocky" images. Using the ℓ_1 norm for the residual term yields very good results if only a small portion of the pixels are contaminated by noise. Figure 5.6(a)–(e) show an original "blocky" image, blurred image with 50% of the pixels contaminated by additive noise, least squares restoration, the LMN solution, and the LAD restoration.

Figures 5.7 and 5.8 show the convergence of the primal-dual interior point algorithm for the LMN and LAD solutions, respectively. Theoretically, the interior point iterations should exhibit superlinear convergence. The experimental results do not seem to confirm the theoretical predictions. This can be explained as follows. The linear systems are approximately solved such that the residual is below a certain level. Because the conditions of the linear systems get worse as the iterates get closer to the solution, the error in the calculated search directions becomes bigger, causing poorer convergence. In fact, for small images, if direct methods are used to solve the linear

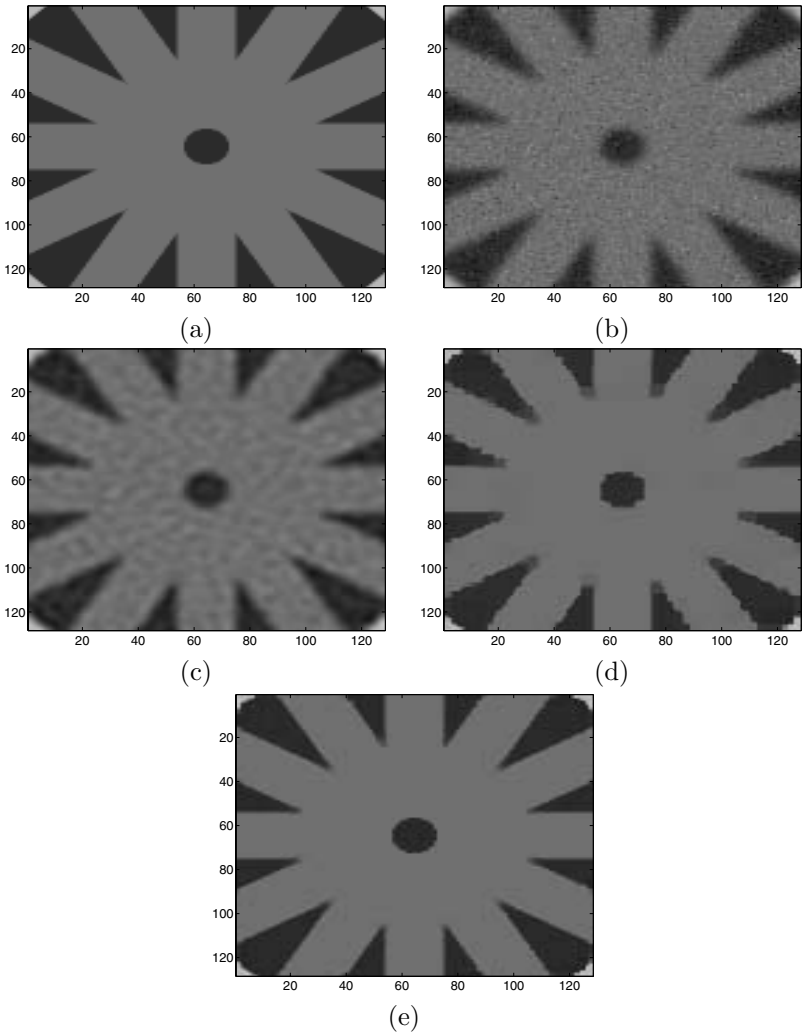


FIG. 5.6. (a) *An original wheel image.* (b) *The observed noisy image with* 50% *of the pixels contaminated by noise.* (c) *The least squares restoration, PSNR* $= 29.25$ *db.* (d) *The LMN solution, PSNR* $= 30.64$ *db.* (e) *The LAD solution, PSNR* $= 35.26$ *db.*

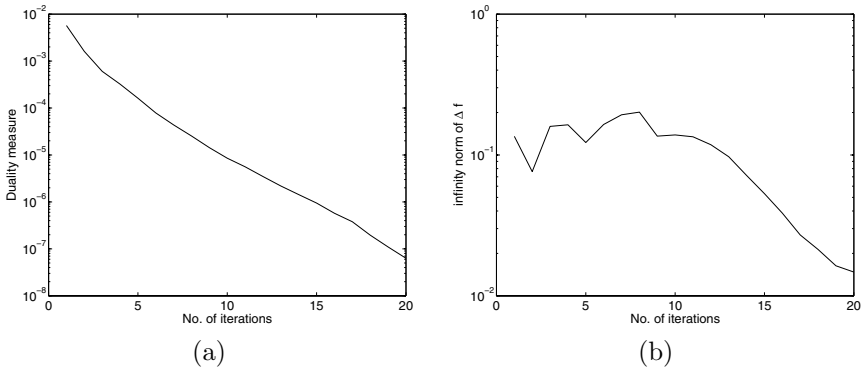
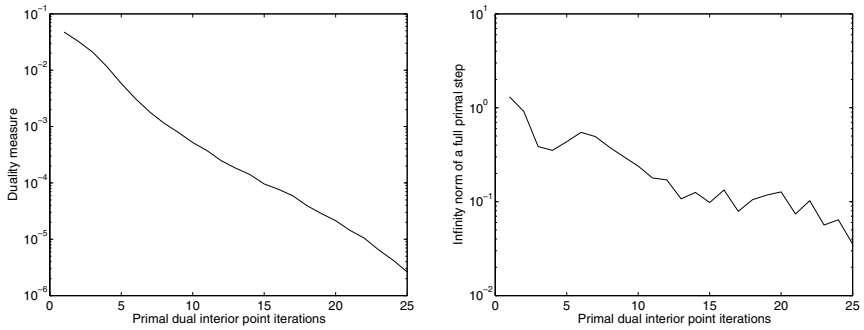


FIG. 5.7. *Convergence of the primal-dual interior point method for the LMN solution:* (a) *the duality measure;* (b) *the infinity norm of* $\Delta \mathbf{f}$.



(a) (b)

FIG. 5.8. *Convergence of the primal-dual interior point method for the LAD solution:* (a) *the duality measure;* (b) *the infinity norm of* $\Delta \mathbf{f}$.

TABLE 5.1

Number of conjugate gradient iterations and the CPU time needed at each interior point iteration for the LAD solution when different preconditioning strategies are used. The preconditioning strategies used include (1) *NOR,* (2.12) *is solved, using a diagonal preconditioner;* (2) *BENZI, the Benzi and Golub preconditioner is applied to* (2.13)*;* (3) *No Pre, plain conjugate gradient is applied to* (2.14)*;* (4) *Diag Pre, diagonal preconditioner is applied to* (2.14)*;* (5) *FSIP*2*, FSIP with* $p = q = 2$ *is applied to* (2.14)*;* (6) *FSIP*3*, FSIP with* $p = q = 3$*;* (7) *FSIP*4*, FSIP with* $p = q = 4$.

PD Itn	NOR	BENZI	No Pre	Diag Pre	FSIP2	FSIP3	FSIP4
10	>3000	295	644	544	48	22	18
	–	161.78	78.68	68.72	15.24	**13.32**	17.08
12	>3000	483	982	804	51	26	21
	–	237.70	125.92	105.53	15.75	**14.14**	18.01
14	>3000	1047	1603	1334	60	31	25
	–	589.68	207.05	175.88	17.42	**15.31**	18.99
16	>3000	1791	2452	1934	77	43	34
	–	969.84	305.14	242.51	20.90	**18.47**	22.14
18	>3000	>2000	>3000	2607	106	54	41
	–	–	–	339.06	25.85	**20.87**	25.11
20	>3000	>2000	>3000	>3000	170	85	58
	–	–	–	–	37.71	**28.18**	28.91

systems, a slightly superlinear convergence is observed.

Table 5.1 shows the number of conjugate gradient iterations and CPU time at each interior point iteration for the LAD solution when different preconditioning strategies are used. The CPU time listed here includes the time for constructing the preconditioners. Table 5.2 shows the same kind of results for the linear systems in the LMN solution. As can be seen, in both cases, the FSIP with proper choices of p and q can significantly speed up the convergence.

We remark that the performance of the FSIP depends on the support of the blurring function and on how fast the function decays in space. The effect of the support of the blurring matrix on the effectiveness of the FSIP is twofold. On one hand, a big support means a relatively dense blurring matrix, therefore, allowing the preconditioner to be relatively dense. On the other hand, a big support usually means a relatively dense preconditioner is needed in order to improve the condition significantly. Table 5.3 shows the performance of the FSIP with a different choice of p and q when the support of the blur is 5×5 (the same Gaussian blur is used). Table 5.4 shows the same kind of information when the support of the blur is 9×9.

TABLE 5.2

Number of conjugate gradient iterations and the CPU time needed at each interior point iteration for the LMN solution when different preconditioners are used. The preconditioning strategies used include (1) *No Pre, plain conjugate gradient is applied to* (3.11)*;* (2) *Diag Pre, diagonal preconditioner is applied to* (3.11)*;* (3) *FSIP*2*, FSIP with* $p = q = 2$ *is applied to* (3.11)*;* (4) *FSIP*3*, FSIP with* $p = q = 3$*;* (5) *FSIP*4*, FSIP with* $p = q = 4$*.*

PD Itn	No Pre	Diag Pre	FSIP2	FSIP3	FSIP4
10	368	180	31	21	15
	45.01	22.94	**6.15**	6.89	10.52
12	618	281	51	32	22
	76.27	35.27	9.71	**9.24**	12.72
14	908	438	85	48	32
	112.37	55.28	16.12	**13.35**	15.86
16	1391	626	139	71	49
	171.36	78.27	25.95	**18.31**	20.84
18	2202	989	225	123	67
	265.25	119.56	40.32	29.99	**26.55**
20	>3000	1814	408	228	120
	–	222.30	72.91	54.15	**41.62**

TABLE 5.3

Number of conjugate gradient iterations and the CPU time needed at each interior point iteration for the LAD solution when different FSIP is used. The support of the blur is 5×5*.*

PD Itn	No Pre	Diag Pre	FSIP2	FSIP3	FSIP4
10	839	669	57	24	14
	79.10	64.95	11.40	**9.39**	11.76
12	1406	1140	96	38	21
	130.74	108.45	17.16	**11.94**	13.40
14	2199	1672	129	52	28
	208.98	164.74	22.31	**14.61**	15.54
16	>3000	2296	175	73	39
	–	222.31	29.55	19.45	**18.64**
18	>3000	>3000	266	106	54
	–	–	43.23	26.57	**22.80**
20	>3000	>3000	378	142	74
	–	–	59.50	32.89	**28.19**

TABLE 5.4

Number of conjugate gradient iterations and the CPU time needed at each interior point iteration for the LAD solution when different FSIP is used. The support of the blur is 9×9*.*

PD Itn	No Pre	Diag Pre	FSIP2	FSIP3	FSIP4
10	1184	578	74	46	30
	198.76	100.03	26.66	**24.41**	25.43
12	1759	940	115	67	46
	286.54	157.77	36.17	**30.23**	31.29
14	2394	1432	175	98	65
	389.36	241.66	49.39	38.63	**38.23**
16	>3000	2093	272	153	97
	–	353.55	71.68	54.44	**48.50**
18	>3000	>3000	409	237	143
	–	–	100.86	76.32	**65.62**
20	>3000	>3000	710	402	236
	–	–	166.37	121.45	**96.83**

TABLE 5.5
Number of conjugate gradient iterations and the CPU time needed at each interior point iteration for the LAD solution when different FSIP is used. The blur is a 7×7 uniform blur.

PD Itn	No Pre	Diag Pre	FSIP2	FSIP3	FSIP4
10	454	346	96	94	88
	57.25	44.83	**23.69**	30.28	38.72
12	731	535	118	113	105
	92.19	69.10	**28.16**	33.76	45.07
14	1241	1022	169	156	141
	158.31	135.20	**37.24**	44.61	55.51
16	1826	1596	250	229	197
	221.76	199.87	**50.20**	61.90	72.77
18	2796	2306	334	287	269
	347.30	295.07	**66.21**	75.45	92.30
20	>3000	>3000	550	464	411
	–	–	**104.73**	117.31	136.25

In general, the faster the blurring function decays in space, the more effective the FSIP is. In that sense, the worst blurring function is a uniform blur. Table 5.5 shows the performance of the FSIP if a 7×7 uniform blur is used.

6. Conclusions. In conclusion, we have proposed efficient algorithms for finding the LAD or LMN solution for the image restoration problem. We formulated these problems as solutions to smooth linear or quadratic programming problems, which are solved by a primal-dual interior point method. The linear system at each interior point iteration is first reduced to a more compact system and then solved by the PCG method. The FSIP is used to speed up the convergence of the conjugate gradient method.

Experimental results indicate that the LAD and LMN solutions have advantages over the least squares solution. We also demonstrated that the FSIP can speed up the convergence considerably if a proper sparsity pattern is chosen.

REFERENCES

[1] N. ABDELMALEK, *Restoration of images with missing high frequency components using quadratic programming*, Appl. Opt., 22 (1983), pp. 2182–2188.
[2] N. ABDELMALEK AND T. KASVAND, *Digital image restoration using quadratic programming*, Appl. Opt., 19 (1980), pp. 3407–3415.
[3] N. ABDELMALEK AND N. OTSU, *Restoration of images with missing high-frequency components by minimizing the $\ell 1$ norm of the solution vector*, Appl. Opt., 24 (1985), pp. 1415–1420.
[4] S. ALLINEY, *Digital filters as absolute norm regularizers*, IEEE Trans. Signal Process., 40 (1992), pp. 1548–1562.
[5] S. ALLINEY, *A property of the minimum vectors of a regularizing functional defined by means of absolute norm*, IEEE Trans. Signal Process., 45 (1997), pp. 913–917.
[6] S. ALLINEY AND S. RUZINSKY, *An algorithm for the minimization of mixed $\ell 1$ and $\ell 2$ norms with application to Bayesian estimation*, IEEE Trans. Signal Process., 42 (1994), pp. 618–627.
[7] M. BENZI AND G. H. GOLUB, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 20–41.
[8] M. BENZI, C. MEYER, AND M. TUMA, *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput., 17 (1996), pp. 1135–1149.
[9] P. BLOOMFIELD AND W. STEIGER, *Least Absolute Deviations: Theory, Applications, and Algorithms*, Birkhäuser, Boston, 1983.
[10] L. CANALES, *L1 Norm Spectral Estimation*, Technical report, Stanford University, Stanford Exploration Project, http://sepwww.stanford.edu/oldreports/sep07/sp07_16.pdf (1975).
[11] A. CHAMBOLLE, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97.

[12] T. Chan and S. Esedoglu, *Aspects of Total Variation Regularized $\ell 1$ Function Approximation*, Technical report, University of California at Los Angeles, 2004.
[13] G. Chavent and K. Kunisch, *Regularization of linear least squares problems by total bounded variation*, ESAIM: Control Optim. Calc. Var., 2 (1997), pp. 359–376.
[14] P. Combettes, *An adaptive level set method for nondifferentiable constrained image recovery*, IEEE Trans. Image Process., 11 (2002), pp. 1295–1304.
[15] A. Dax, *A row relaxation method for large $\ell 1$ problems*, Linear Algebra Appl., 156 (1991), pp. 793–818.
[16] A. Dax, *On regularized least norm problems*, SIAM J. Optim., 2 (1992), pp. 602–618.
[17] A. Dax, *On row-relaxation methods for large constrained least squares problems*, SIAM J. Sci. Comput., 14 (1993), pp. 570–584.
[18] A. Dax, *A row relaxation method for large ℓ_p least norm problems*, Numer. Linear Algebra Appl., 1 (1994), pp. 247–263.
[19] D. Etter, D. Kuncicky, and D. Hull, *Introduction to Matlab* 6, Prentice-Hall, Upper Saddle River, NJ, 2002.
[20] H. Fu and J. Barlow, *A regularized structured total least squares algorithm for high resolution image reconstruction*, Linear Algebra Appl., 391 (2004), pp. 75–98.
[21] A. Guitton and D. J. Verschuur, *Adaptive subtraction of multiples using the $\ell 1$-norm*, Geophys. Prospecting, 52 (2004), pp. 1–27.
[22] M. Hanke, J. Nagy, and C. Vogel, *Quasi-Newton approach to nonnegative image restorations*, Linear Algebra Appl., 316 (2000), pp. 223–236.
[23] M. Hintermüller and K. Kunisch, *Total bounded variation regularization as a bilaterally constrained optimization problem*, SIAM J. Appl. Math., 64 (2004), pp. 1311–1333.
[24] K. Ito and K. Kunisch, *An active set strategy based on the augmented Lagrangian formulation for image restoration*, ESAIM: Math. Model. Numer. Anal., 33 (1999), pp. 1–21.
[25] T. Kärkkäinen, K. Kunisch, and K. Majava, *Denoising of smooth images using L^1-fitting*, Computing, 74 (2005), pp. 353–376.
[26] Q. Ke and T. Kanade, *Robust Subspace Computation Using $\ell 1$ Norm*, Technical report, School of Computer Science, Carnegie Mellon University, CMU-CS-03-172, 2003.
[27] L. Kolotilina and A. Yeremin, *Factorized sparse approximate inverse preconditionings* I. *Theory*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 45–58.
[28] S. Kuo and R. Mammone, *Image restoration by convex projections using adaptive constraints and the $\ell 1$ norm*, IEEE Trans. Signal Process., 40 (1992), pp. 1159–1168.
[29] Y. Li and F. Santosa, *A computational algorithm for minimizing total variation in image restoration*, IEEE Trans. Image Process., 6 (1996), pp. 987–995.
[30] F.-R. Lin, M. K. Ng, and W.-K. Ching, *Factorized banded inverse preconditioners for matrices with Toeplitz structure*, SIAM J. Sci. Comput., 26 (2005), pp. 1852–1870.
[31] C. van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, 1992.
[32] R. Mammone and G. Eichmann, *Superresolving image restoration using linear programming*, Appl. Opt., 21 (1983), pp. 496–501.
[33] N. Mastronardi, P. Lemmerling, A. Kalsi, D. O'Leary, and S. van Huffel, *Implementation of regularized structured total least squares algorithms for blind image deblurring*, Linear Algebra Appl., 391 (2004), pp. 203–221.
[34] V. Mesarović, N. Galatsanos, and A. Katsaggelos, *Regularized constrained total least squares image restoration*, IEEE Trans. Image Process., 4 (1995), pp. 1096–1108.
[35] T. Miyashita, *Super-resolved image restoration of holographic images by $\ell 1$-norm minimization with clutter rejection*, Acoustical Imaging, 19 (1992), pp. 77–82.
[36] M. Ng, R. Chan, and W. Tang, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–866.
[37] M. K. Ng, R. J. Plemmons, and F. Pimentel, *A new approach to constrained total least squares image restoration*, Linear Algebra Appl., 316 (2000), pp. 237–258.
[38] M. Nikolova, *Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers*, SIAM J. Numer. Anal., 40 (2002), pp. 965–994.
[39] M. Nikolova, *A variational approach to remove outliers and impulse noise*, J. Math. Imaging Vision, 20 (2004), pp. 99–120.
[40] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, New York, 1999.
[41] C. Paige and M. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[42] A. Pruessner and D. O'Leary, *Blind deconvolution using a regularized structured total least norm algorithm*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 1018–1037.
[43] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.

[44] S. Ruzinsky, *Sequential Least Absolute Deviation Estimation of Autoregressive Parameters*, Ph.D. thesis, Illinois Institute of Technology, Chicago, IL, 1989.
[45] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[46] M. Saunders, *Interior Methods for Optimization with Application to Maximum Entropy Problems*, http://sccm.stanford.edu/csri/Saunders.pdf.
[47] W. Tang, *Toward an effective sparse approximate inverse preconditioner*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 970–986.
[48] A. Tikhonov and V. Arsenin, *Solution of Ill-Posed Problems*, Winston, Washington, DC, 1977.
[49] S. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.