Electronic Transactions on Numerical Analysis.
Volume 55, pp. 169–186, 2022.

ISSN 1068–9613.
DOI: 10.1553/etna_vol55s169

ETNA
Kent State University and
Johann Radon Institute (RICAM)

GRAPH LAPLACIAN FOR IMAGE DEBLURRING*

DAVIDE BIANCHI†, ALESSANDRO BUCCINI‡, MARCO DONATELLI§, AND EMMA RANDAZZO§

Abstract. Image deblurring is a relevant problem in many fields of science and engineering. To solve this problem, many different approaches have been proposed, and, among the various methods, variational ones are extremely popular. These approaches substitute the original problem with a minimization problem where the functional is composed of two terms, a data fidelity term and a regularization term. In this paper we propose, in the classical non-negative constrained ℓ^2-ℓ^1 minimization framework, the use of the graph Laplacian as regularization operator. Firstly, we describe how to construct the graph Laplacian from the observed noisy and blurred image. Once the graph Laplacian has been built, we efficiently solve the proposed minimization problem by splitting the convolution operator and the graph Laplacian by the Alternating Direction Multiplier Method (ADMM). Some selected numerical examples show the good performances of the proposed algorithm.

Key words. image deblurring, graph Laplacian, ℓ^2-ℓ^1 regularization

AMS subject classifications. 65R32, 65K10, 65F22

1. Introduction.

We are concerned with the problem of space-invariant image deblurring, which can be modeled as a linear system of equations

$$A\mathbf{x} = \mathbf{b},$$

where $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{b} \in \mathbb{R}^M$ are samplings of the unknown image that we wish to recover and of the blurred image, respectively. The matrix $A \in \mathbb{R}^{M\times N}$ is structured (see below) and severely ill-conditioned, i.e., its singular values decay rapidly to 0 with no significant gap; see, e.g., [33] for more details. The discretization process, along with measurement errors, introduces some perturbations in the data, namely $\boldsymbol{\eta} \in \mathbb{R}^M$, often referred to as noise, leading to the system

$$A\mathbf{x} = \mathbf{b} + \boldsymbol{\eta} = \mathbf{b}^\delta.$$

Since, in general, $\boldsymbol{\eta} \notin \mathcal{R}(A)$, we reformulate the system above as a least-squares problem

$$\arg\min_{\mathbf{x}} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|, \tag{1.1}$$

where $\|\cdot\|$ denotes the Euclidean norm. Let $A^\dagger$ denote the Moore-Penrose pseudo-inverse of A. The naive solution of (1.1), $A^\dagger\mathbf{b}^\delta$, is usually a poor approximation of the desired solution $\mathbf{x}^\dagger = A^\dagger\mathbf{b}$; see, e.g., [24, 32]. Since A is severely ill-conditioned, the solution of (1.1) is extremely sensitive to the presence of noise in the data. To compute an accurate approximation of $\mathbf{x}^\dagger$ we resort to regularization methods. These methods aim at reducing the sensitivity of

*Received May 2, 2021. Accepted November 3, 2021. Published online on December 3, 2021. Recommended by Stefan Kindermann. A.B. and M.D. are members of the GNCS-INdAM group. D.B. is member of the GNAMPA-INdAM group. A.B. research is partially supported by the Regione Autonoma della Sardegna research project "Algorithms and Models for Imaging Science [AMIS]" (RASSR57257, intervento finanziato con risorse FSC 2014–2020-Patto per lo Sviluppo della Regione Sardegna).

†School of Science Harbin Institute of Technology (Shenzhen) 518055 Shenzhen, China (bianchi@hit.edu.cn)

‡Department of Mathematics and Computer Science University of Cagliari 09124 Cagliari, Italy (alessandro.buccini@unica.it)

§Department of Science and High Technology University of Insubria 22100 Como, Italy ({marco.donatelli, erandazzo1}@uninsubria.it)

ETNA
Kent State University and
Johann Radon Institute (RICAM)

the problem with respect to the presence of noise. Probably the most popular regularization method is Tikhonov regularization, where the original minimization problem is substituted by

$$\arg\min_{\mathbf{x}} \left\| A\mathbf{x} - \mathbf{b}^{\delta} \right\|^2 + \mu \left\| L\mathbf{x} \right\|^2, \tag{1.2}$$

where $L \in \mathbb{R}^{p \times n}$ is chosen so that $\mathcal{N}(A) \cap \mathcal{N}(L) = \{\mathbf{0}\}$. The matrix L is the so-called regularization operator and its role is to enforce some a-priori knowledge on the reconstruction. If A is the discretization of an integral operator, then L is usually chosen to be a discretization of either the first or the second derivative; see, e.g., [23].

The formulation (1.2) can be extended to a general ℓ_p-norm by considering

$$\arg\min_{\mathbf{x}} \left\| A\mathbf{x} - \mathbf{b}^{\delta} \right\|^2 + \mu \left\| L\mathbf{x} \right\|_p^p, \tag{1.3}$$

where $\|\mathbf{x}\|_p^p = \sum_{i=1}^{n} |x_i|^p$ for $p > 0$; see, e.g., [14, 21, 25, 34, 38]. Note that for $p < 1$ the function $\mathbf{x} \mapsto \|\mathbf{x}\|_p$ is not a norm.

In this paper, we propose to use an appropriately constructed graph Laplacian for L in (1.3) with $p = 1$. Therefore, our minimization problem is of the form

$$\arg\min_{\mathbf{x} \geq 0} \left\| A\mathbf{x} - \mathbf{b}^{\delta} \right\|^2 + \mu \left\| L\mathbf{x} \right\|_1. \tag{1.4}$$

Since we are considering imaging problems, we know that the solution cannot have negative entries; therefore we have introduced non-negativity constraints; see, e.g., [2, 22, 42].

Recently, the graph Laplacian of a network, built from an approximation of $\mathbf{x}^{\dagger}$, has been proposed as regularization operator for image denoising, i.e., for problems of the form (1.1) with $A = I$, where I denotes the identity matrix; see. e.g., [36, 37, 39, 41, 43, 46, 50]. For the case of image denoising, the approximation of $\mathbf{x}^{\dagger}$ used for the construction of the graph is usually given by the noisy data $\mathbf{b}^{\delta}$.

In this paper, we consider the more general case of image deblurring. In such a case, $\mathbf{b}^{\delta}$ is no longer a good approximation of $\mathbf{x}^{\dagger}$, thus we propose a two-step procedure to first compute an appropriate graph Laplacian and then solve (1.4). The construction of the graph Laplacian that we propose is completely automatic and does not need any intervention from the user. The image used to construct the graph Laplacian is cheaply computed by Tikhonov regularization with an automatic estimation of the parameter μ. Note that the proposed regularization operator could be considered for other regularization methods as well, for instance, for solving equation (1.3) with $p \neq 1$.

A second contribution of this paper is the efficient solution of the minimization problem (1.4). From a computational point of view, the convolution operator A cannot be easily combined with the graph Laplacian L because the matrix A can be diagonalized by a Fast Fourier transform (`fft`), while this is not the case for L, which, however, is sparse. Therefore, we solve (1.4) by the Alternating Direction Multiplier Method (ADMM) [7] such that the inner steps can be computed by `fft` and Krylov methods. Finally, in some selected numerical examples we compare the reconstructed images obtained by our problem (1.4) with the graph Laplacian and some recently proposed methods including the standard Total Variation (TV); see [45]. The result is that our proposal can lead to substantial improvements in the quality of the reconstructed images.

This paper is structured as follows: in Section 2 we recall the definition of the Laplacian of a given graph, and we construct the one that we use in the following. Section 3 presents our algorithmic proposal for the solution of (1.4) and Section 4 contains some selected numerical experiments. Finally, we draw some conclusions in Section 5.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Notation. Discretized images consist of the union of several pixels in the plane and therefore are well represented by nonnegative two-dimensional discrete functions, i.e., $\mathbf{x} : \mathbb{R}^{n_1 \times n_2} \to [0, +\infty)$, $\mathbf{x}(i_1, i_2) = x_{i_1,i_2} \in \mathbb{R}_+$, for $i_1 = 1, \dots, n_1$, $i_2 = 1, \dots, n_2$. For our purposes we reorder the pixels so that the image is represented by a one-dimensional discrete function. For the sake of notational simplicity, we fix $n_1 = n_2 = n = \sqrt{N}$ and consider the lexicographic one-dimensional ordering, that is, $(i_1, i_2) =: i < j := (j_1, j_2)$ if $i_1 < j_1$ or $i_1 = j_1$ and $i_2 < j_2$. With this choice, an image reads as $\mathbf{x} : \mathbb{R}^N \to [0, +\infty)$, $\mathbf{x}(i) = x_i \in \mathbb{R}_+$, for $i = 1, \dots, N$, and we say that $\mathbf{x}$ is the vectorization of a square image.

2. Construction of the graph Laplacian. In this section, we first describe how to construct the Laplacian of a given weighted graph. Then we show how to build an appropriate graph, i.e., a graph whose Laplacian is a "good" regularization operator given a good approximation of the exact image $\mathbf{x}^\dagger$. Finally, we provide an algorithm for constructing our matrix L for the problem (1.4).

Given a countable measure space $(\mathcal{V}, \nu)$, where ν is a positive measure. A symmetric non-negative function $\omega : \mathcal{V} \times \mathcal{V} \to [0, +\infty)$ with zero diagonal is called an undirected *graph* on $\mathcal{V}$. The elements i, j of the set $\mathcal{V}$ are called *vertices*, and two vertices are *connected* if $\omega(i, j) > 0$. The positive value $w(i, j)$ is called *weight* associated to the *edge* $\{i, j\}$; for an introduction to graph theory we refer to [35]. If $\mathcal{V}$ is a finite set of N elements, then the graph ω can be uniquely represented, up to permutations, by the *adjacency matrix* $\Omega \in \mathbb{R}^{N \times N}$,

$$\Omega_{i,j} := \omega(i, j).$$

The linear operator $L_\omega : C(\mathcal{V}) \to C(\mathcal{V})$ acting on the space $C(\mathcal{V}) := \{\mathbf{x} : \mathcal{V} \to \mathbb{R}\} \simeq \mathbb{R}^N$ via

$$L_\omega \mathbf{x}(i) := \frac{1}{\nu(i)} \sum_j \omega(i, j) \left(\mathbf{x}(i) - \mathbf{x}(j)\right)$$

is the graph Laplacian on $C(\mathcal{V})$ associated to the graph ω. It is a symmetric operator with respect to the inner product

$$\langle \mathbf{x}, \mathbf{y} \rangle := \sum_i \mathbf{x}(i)\mathbf{y}(i)\nu(i).$$

In many applications, a quite standard choice for the measure ν is the *degree* function $\nu = \deg$ defined by

$$\deg(i) := \sum_j \omega(i, j)$$

since it measures the whole intensity of the weights associated to the vertex i. Clearly, this choice makes L_ω non-symmetric with respect to the standard Euclidean inner product. A good compromise is to choose the homogeneous measure associated to the Frobenius norm of Ω, i.e., $\nu(i) \equiv \|\Omega\|_F$. Let us observe that, writing D as the diagonal matrix such that $(D)_{i,i} = \deg(i)$, then it is easy to verify that

$$\frac{1}{N}\|D\|_1 \le \|\Omega\|_F \le \|D\|_1,$$

where we denote by $\|D\|_1$ the 1-norm of the vector obtained by extracting the diagonal of D.

Henceforth, we will assume $\nu = \|\Omega\|_F$. In matrix form, the graph Laplacian reads

$$L_\omega = \frac{D - \Omega}{\|\Omega\|_F}.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

We wish to construct a graph ω so that L_ω can be used in (1.4). In principle, we would like to construct ω such that

$$\left\|L_\omega \mathbf{x}^\dagger\right\| \approx 0.$$

To this aim, let $\mathbf{x}^*$ be a good approximation of $\mathbf{x}^\dagger$. Intuitively, the graph is constructed as follows: the nodes are the pixels of $\mathbf{x}^*$ and we connect two pixels if they are "spatially" close enough. The weight of their connection depends on how similar their values are. In particular, we give a strong connection to pixels that have similar values. We introduce a parameter R that determines how large the neighborhood is that we consider for each pixel and a parameter σ that determines how strong the connection between two close pixels should be.

Define ω as the weighted and undirected graph on $\mathcal{V}$ whose nodes are the pixels of $\mathbf{x}^*$ and where the weights are defined by

$$\omega(i,j) = \begin{cases} \mathrm{e}^{-(\mathbf{x}^*(i)-\mathbf{x}^*(j))^2/\sigma} & \text{if } i \neq j \text{ and } \|r(i)-r(j)\|_\infty \leq R, \\ 0 & \text{otherwise.} \end{cases}$$

Here $r(i) = (i_1, i_2)$ is the two-dimensional index of the pixel $\mathbf{x}^*(i)$, and $\sigma > 0$ and $R \in \mathbb{N}$ are user-defined parameters. The choice of the Gaussian function that appears in the definition of ω is motivated by its relation to the heat-kernel. For the motivation behind this common choice (see, e.g., [20, 44] and the references therein), we refer to [47, Section 3.1.2]. We recall that if $r(i) = (i_1, i_2), r(j) = (j_1, j_2)$, then $\|r(i) - r(j)\|_\infty = \max\{|i_1 - j_1|, |i_2 - j_2|\}$.

The construction of this graph and consequently of the graph Laplacian, in turn, depends on the construction of an appropriate approximate solution $\mathbf{x}^*$. As we show in Section 4, if we could choose $\mathbf{x}^* = \mathbf{x}^\dagger$, we would obtain an almost optimal result. However, this is not possible in realistic scenarios. Therefore, we wish to provide a practical way to determine a good enough $\mathbf{x}^*$ in a totally automatic way.

To compute $\mathbf{x}^*$ we propose to solve (1.2) with a regularization operator defined as follows. Let $L_1 \in \mathbb{R}^{n\times n}$ be

$$L_1 = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ 1 & & & & -1 \end{bmatrix},$$

i.e., L_1 is a discretization of the first derivative with periodic boundary conditions (BCs), and let I_n be the identity matrix of order n. Then we define L_{TV} by

$$L_{\mathrm{TV}} = \begin{bmatrix} L_1 \otimes I_n \\ I_n \otimes L_1 \end{bmatrix} \in \mathbb{R}^{2N\times N}. \tag{2.1}$$

This choice is inspired by the well-known Total Variation approach described, e.g., in [45]. Intuitively, we are looking for a solution whose gradient has small norm, i.e., one that is regular enough; see, e.g., [23] for more details.

Note that L_{TV} is an extremely sparse matrix formed by two Block Circulant with Circulant Blocks (BCCB) matrices stacked one above the other. Therefore, matrix-vector products involving L_{TV} can be performed extremely cheaply (in particular, the flop count is $O(N)$) and $L_{\mathrm{TV}}^T L_{\mathrm{TV}}$ is a BCCB matrix. We exploit the latter property below. We recall that a circulant

ETNA
Kent State University and
Johann Radon Institute (RICAM)

matrix is a matrix of the form

$$C = \begin{bmatrix} c_1 & c_2 & c_3 & \dots & c_n \\ c_n & c_1 & c_2 & \dots & c_{n-1} \\ c_{n-1} & c_n & c_1 & \dots & c_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_2 & c_3 & c_4 & \dots & c_1 \end{bmatrix},$$

i.e., a matrix such that the ith row is obtained by shifting the $(i-1)$st entry, imposing periodic boundary conditions.

To simplify the computations we impose periodic BCs for the matrix A. Thanks to this choice, A is a BCCB matrix; see [33] for more details. We recall that BCCB matrices are diagonalized by the two-dimensional Fourier matrix. Let $F_1 \in \mathbb{R}^{n\times n}$ be the Fourier matrix, i.e., $(F_1)_{j,k} = \mathrm{e}^{2\pi\iota(j-1)(k-1)/n}$ with $\iota^2 = -1$. Then the two-dimensional Fourier matrix is defined by $F = F_1 \otimes F_1$. Note that matrix-vector products with F and its inverse F^* can be performed in $O(N \log N)$ flops with the aid of the `fft` and `ifft` algorithms; see, e.g., [33].

As discussed above, we wish to solve (1.2) with $L = L_{\mathrm{TV}}$, i.e., we wish to solve a problem of the form

$$\mathbf{x}_\mu = \arg\min_{\mathbf{x}} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|^2 + \mu \left\| L_{\mathrm{TV}}\mathbf{x} \right\|^2$$

for a certain $\mu > 0$. Thanks to the structure of A and L_{TV} this can be solved cheaply for any μ. We can write

$$A = F^*\Sigma F \quad \text{and} \quad L_{\mathrm{TV}} = \begin{bmatrix} F^*\Lambda_x F \\ F^*\Lambda_y F \end{bmatrix}, \tag{2.2}$$

where Σ, Λ_x, and Λ_y are diagonal matrices whose diagonal entries are the eigenvalues of A, $L_1 \otimes I_n$, and $I_n \otimes L_1$, respectively. We recall that the eigenvalues of a BCCB matrix C can be computed by $F\mathbf{c}_1$, where $\mathbf{c}_1$ is the first column of C. Assuming that $\mathcal{N}(A) \cap \mathcal{N}(L_{\mathrm{TV}}) = \{\mathbf{0}\}$, we have that

$$\begin{aligned} \mathbf{x}_\mu &= (A^TA + \mu L_{\mathrm{TV}}^T L_{\mathrm{TV}})^{-1} A^T \mathbf{b}^\delta \\ &= \left(F^*\Sigma^* F F^* \Sigma F + \mu \begin{bmatrix} F^*\Lambda_x^* F & F^*\Lambda_y^* F \end{bmatrix} \begin{bmatrix} F^*\Lambda_x F \\ F^*\Lambda_y F \end{bmatrix} \right)^{-1} F^*\Sigma^* F \mathbf{b}^\delta \\ &= \left(F^*\Sigma^*\Sigma F + \mu F^*(\Lambda_x^*\Lambda_x + \Lambda_y^*\Lambda_y)F \right)^{-1} F^*\Sigma^* F \mathbf{b}^\delta \\ &= F^* \left(\Sigma^*\Sigma + \mu(\Lambda_x^*\Lambda_x + \Lambda_y^*\Lambda_y) \right)^{-1} \Sigma^* F \mathbf{b}^\delta, \end{aligned} \tag{2.3}$$

where the matrix to be inverted is a diagonal matrix. Therefore, $\mathbf{x}_\mu$ can be computed with small effort for any μ.

We now wish to determine the parameter μ in an automatic way. We employ the Generalized Cross Validation (GCV); see, e.g., [31]. The GCV parameter μ_{GCV} is the minimizer of the function

$$G(\mu) = \frac{\left\| A\mathbf{x}_\mu - \mathbf{b}^\delta \right\|^2}{\mathrm{trace}(I - A(A^TA + \mu L_{\mathrm{TV}}^T L_{\mathrm{TV}})^{-1} A^T)^2},$$

i.e., $\mu_{\mathrm{GCV}} = \arg\min_\mu G(\mu)$. Intuitively, the GCV parameter is chosen starting from the idea that if a data point is omitted in $\mathbf{b}^\delta$, then a good reconstruction should be able to predict the missing pixel.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Given the decomposition (2.2), the value of $G(\mu)$ can be computed in a straightforward way. Introduce the following notation

$$r_\mu = \left\|A\mathbf{x}_\mu - \mathbf{b}^\delta\right\| \quad \text{and} \quad t_\mu = \operatorname{trace}(I - A(A^TA + \mu L_{\mathrm{TV}}^T L_{\mathrm{TV}})^{-1}A^T),$$

thus, $G(\mu) = r_\mu^2/t_\mu^2$. Using the spectral decomposition of A we have

$$\begin{aligned} r_\mu &= \left\|A\mathbf{x}_\mu - \mathbf{b}^\delta\right\| = \left\|F^*\Sigma F F^* \left(\Sigma^*\Sigma + \mu(\Lambda_x^*\Lambda_x + \Lambda_y^*\Lambda_y)\right)^{-1}\Sigma^* F\mathbf{b}^\delta - \mathbf{b}^\delta\right\| \\ &= \left\|\Sigma\left(\Sigma^*\Sigma + \mu(\Lambda_x^*\Lambda_x + \Lambda_y^*\Lambda_y)\right)^{-1}\Sigma^* F\mathbf{b}^\delta - F\mathbf{b}^\delta\right\| \\ &= \left\|(\Sigma\Sigma^*\left(\Sigma^*\Sigma + \mu(\Lambda_x^*\Lambda_x + \Lambda_y^*\Lambda_y)\right)^{-1} - I)\widehat{\mathbf{b}^\delta}\right\|, \end{aligned}$$

where $\widehat{\mathbf{b}^\delta} = F\mathbf{b}^\delta$. We now move to the computation of t_μ

$$\begin{aligned} t_\mu &= \operatorname{trace}(I - A(A^TA + \mu L_{\mathrm{TV}}^T L_{\mathrm{TV}})^{-1}A^T) \\ &= \operatorname{trace}(I - F^*\Sigma(\Sigma^*\Sigma + \mu\left(\Lambda_x^*\Lambda_x + \Lambda_y^*\Lambda_y\right))^{-1}\Sigma^* F) \\ &= \operatorname{trace}(I - \Sigma\Sigma^*(\Sigma^*\Sigma + \mu\left(\Lambda_x^*\Lambda_x + \Lambda_y^*\Lambda_y\right))^{-1}). \end{aligned}$$

We observe that once the decompositions (2.2) and $\widehat{\mathbf{b}^\delta}$ have been computed, the evaluation of $G(\mu)$ can be done in $O(N)$ flops. This allows for an extremely fast determination of μ_{GCV} by a standard minimization algorithm. Finally, we select as $\mathbf{x}^*$ the solution of the minimization problem

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \left\|A\mathbf{x} - \mathbf{b}^\delta\right\|^2 + \mu_{\mathrm{GCV}} \left\|L_{\mathrm{TV}}\mathbf{x}\right\|^2$$

computed by

$$\mathbf{x}^* = F^*\left(\Sigma^*\Sigma + \mu_{\mathrm{GCV}}(\Lambda_x^*\Lambda_x + \Lambda_y^*\Lambda_y)\right)^{-1}\Sigma^*\widehat{\mathbf{b}^\delta}$$

thanks to equation (2.3).

We summarize the procedure to construct L_ω in Algorithm 1.

REMARK 2.1. If A is constructed with BCs different from the periodic ones, then it is still possible to compute a fairly accurate approximation of $G(\mu)$ using Krylov subspace methods; see [16, 26, 27].

3. Graph Laplacian deblurring. We now describe the nonlinear model that we employ to compute an approximate solution of (1.1). We consider the graph Laplacian L_ω constructed by Algorithm 1, and we use it in (1.4). Therefore, we wish to solve the following minimization problem:

$$\arg\min_{\mathbf{x}\geq 0} \frac{1}{2}\left\|A\mathbf{x} - \mathbf{b}^\delta\right\|^2 + \mu\left\|L_\omega\mathbf{x}\right\|_1. \tag{3.1}$$

To solve this problem we use ADMM; see, e.g., [7] for a recent review. We use this method since it allows us to decouple the ℓ^2- and ℓ^1-norms as well as the matrices A and L_ω. The latter point is extremely relevant since, as we discuss below, both matrices have exploitable structures, but they are quite different, and, if coupled, this can be difficult to exploit.

We first reformulate (3.1) into an equivalent form

$$\arg\min_{\mathbf{x},\mathbf{y},\mathbf{w},\mathbf{z}} \left\{\frac{1}{2}\left\|A\mathbf{x} - \mathbf{b}^\delta\right\|^2 + \mu\left\|\mathbf{z}\right\|_1 + \iota_0(\mathbf{w}),\ \text{s.t. } \mathbf{x} = \mathbf{y}, \mathbf{x} = \mathbf{w}, \mathbf{z} = L_\omega\mathbf{y},\right\}, \tag{3.2}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 1 Construction of L_ω for image deblurring.

Input: $A \in \mathbb{R}^{N\times N}$, $\mathbf{b}^\delta \in \mathbb{R}^N$, $R > 0$, $\sigma > 0$
Output: $L_\omega \in \mathbb{R}^{n\times n}$
Construct L_{TV} as defined in (2.1)
$\Sigma = \operatorname{diag}\left(F(A_{(:,1)})\right)$
$\Lambda_x = \operatorname{diag}\left(F((L_{\mathrm{TV}})_{(:,1:N)})\right)$
$\Lambda_y = \operatorname{diag}\left(F((L_{\mathrm{TV}})_{(:,N+1:2N)})\right)$
$\widehat{\mathbf{b}^\delta} = F\mathbf{b}^\delta$
$\mu_{\mathrm{GCV}} = \arg\min_\mu \frac{r_\mu^2}{t_\mu^2}$, where

$$\begin{cases} r_\mu = \left\| (\Sigma\Sigma^* \left(\Sigma^*\Sigma + \mu(\Lambda_x^*\Lambda_x + \Lambda_y^*\Lambda_y)\right)^{-1} - I)\widehat{\mathbf{b}^\delta} \right\| \\ t_\mu = \operatorname{trace}(I - \Sigma\Sigma^*(\Sigma^*\Sigma + \mu\left(\Lambda_x^*\Lambda_x + \Lambda_y^*\Lambda_y\right))^{-1}) \end{cases}$$

$\mathbf{x}^* = F^* \left(\Sigma^*\Sigma + \mu_{\mathrm{GCV}}(\Lambda_x^*\Lambda_x + \Lambda_y^*\Lambda_y)\right)^{-1} \Sigma^*\widehat{\mathbf{b}^\delta}$
Construct $\Omega \in \mathbb{R}^{n\times n}$ as

$$\omega(i,j) = \begin{cases} \mathrm{e}^{-(x^*(i)-x^*(j))^2/\sigma} & \text{if } i \neq j \text{ and } \|r(i) - r(j)\|_\infty \leq R, \\ 0 & \text{else,} \end{cases}$$

where $r(i) = (i_1, i_2)$ is the two-dimensional index of the pixel $\mathbf{x}^*(i)$
$D = \operatorname{diag}\{\sum_{j=1}^n (\Omega)_{(i,j)}\}$
$L_\omega = \frac{D-\Omega}{\|\Omega\|_F}$

or equivalently,

$$\begin{aligned} &\arg\min_{\mathbf{x},\mathbf{y},\mathbf{w},\mathbf{z}} \frac{1}{2}\left\|A\mathbf{x} - \mathbf{b}^\delta\right\|^2 + \mu\|\mathbf{z}\|_1 + \iota_0(\mathbf{w}) \\ &\text{s.t.} \begin{bmatrix} I & O \\ O & I \\ I & O \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} - \begin{bmatrix} I & O \\ L_\omega & O \\ O & I \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{w} \end{bmatrix} = \mathbf{0}, \end{aligned} \tag{3.3}$$

where O and $\mathbf{0}$ denote the zero matrix and the zero vector, respectively, and ι_0 is the indicator function of the nonnegative cone, i.e.,

$$\iota_0(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \geq 0, \\ +\infty & \text{otherwise.} \end{cases}$$

We can construct the augmented Lagrangian of (3.3) by

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{x},\mathbf{y},\mathbf{w},\mathbf{z};\boldsymbol{\lambda}) =& \frac{1}{2}\left\|A\mathbf{x} - \mathbf{b}^\delta\right\|^2 + \mu\|\mathbf{z}\|_1 + \iota_0(\mathbf{w}) \\ &+ \left\langle \boldsymbol{\lambda}, \begin{bmatrix} I & O \\ O & I \\ I & O \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} - \begin{bmatrix} I & O \\ L_\omega & O \\ O & I \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{w} \end{bmatrix} \right\rangle \\ &+ \frac{\rho}{2}\left\| \begin{bmatrix} I & O \\ O & I \\ I & O \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} - \begin{bmatrix} I & O \\ L_\omega & O \\ O & I \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{w} \end{bmatrix} \right\|^2, \end{aligned}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where $\rho > 0$ is a fixed parameter and $\boldsymbol{\lambda} \in \mathbb{R}^{3N}$ is the Lagrangian multiplier. Applying ADMM we get the iterations

$$\begin{cases} \begin{bmatrix} \mathbf{x}^{(k+1)} \\ \mathbf{z}^{(k+1)} \end{bmatrix} = \arg\min_{\mathbf{x},\mathbf{z}} \mathcal{L}_\rho(\mathbf{x}, \mathbf{y}^{(k)}, \mathbf{w}^{(k)}, \mathbf{z}; \boldsymbol{\lambda}^{(k)}), \\ \begin{bmatrix} \mathbf{y}^{(k+1)} \\ \mathbf{w}^{(k+1)} \end{bmatrix} = \arg\min_{\mathbf{y},\mathbf{w}} \mathcal{L}_\rho(\mathbf{x}^{(k+1)}, \mathbf{y}, \mathbf{w}, \mathbf{z}^{(k+1)}; \boldsymbol{\lambda}^{(k)}), \\ \boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \rho \left(\begin{bmatrix} I & O \\ O & I \\ I & O \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(k+1)} \\ \mathbf{z}^{(k+1)} \end{bmatrix} - \begin{bmatrix} I & O \\ L_\omega & O \\ O & I \end{bmatrix} \begin{bmatrix} \mathbf{y}^{(k+1)} \\ \mathbf{w}^{(k+1)} \end{bmatrix} \right). \end{cases}$$

We can write $\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ \boldsymbol{\lambda}_3 \end{bmatrix}$ with $\boldsymbol{\lambda}_j \in \mathbb{R}^N$, for $j = 1, 2, 3$. Therefore, the minimization problems above decouples, and we obtain

$$\begin{cases} \mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} \frac{1}{2} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|^2 + \left\langle \begin{bmatrix} \boldsymbol{\lambda}_1^{(k)} \\ \boldsymbol{\lambda}_3^{(k)} \end{bmatrix}, \begin{bmatrix} \mathbf{x} - \mathbf{y}^{(k)} \\ \mathbf{x} - \mathbf{w}^{(k)} \end{bmatrix} \right\rangle + \frac{\rho}{2} \left\| \begin{bmatrix} \mathbf{x} - \mathbf{y}^{(k)} \\ \mathbf{x} - \mathbf{w}^{(k)} \end{bmatrix} \right\|^2, \\ \mathbf{z}^{(k+1)} = \arg\min_{\mathbf{z}} \mu \left\| \mathbf{z} \right\|_1 + \left\langle \boldsymbol{\lambda}_2^{(k)}, \mathbf{z} - L_\omega \mathbf{y}^{(k)} \right\rangle + \frac{\rho}{2} \left\| \mathbf{z} - L_\omega \mathbf{y}^{(k)} \right\|^2, \\ \mathbf{y}^{(k+1)} = \arg\min_{\mathbf{y}} \left\langle \begin{bmatrix} \boldsymbol{\lambda}_1^{(k)} \\ \boldsymbol{\lambda}_2^{(k)} \end{bmatrix}, \begin{bmatrix} \mathbf{x}^{(k+1)} - \mathbf{y} \\ \mathbf{z}^{(k+1)} - L_\omega \mathbf{y} \end{bmatrix} \right\rangle + \frac{\rho}{2} \left\| \begin{bmatrix} \mathbf{x}^{(k+1)} - \mathbf{y} \\ \mathbf{z}^{(k+1)} - L_\omega \mathbf{y} \end{bmatrix} \right\|^2, \\ \mathbf{w}^{(k+1)} = \arg\min_{\mathbf{w}} \iota_0(\mathbf{w}) + \left\langle \boldsymbol{\lambda}_3^{(k)}, \mathbf{x}^{(k+1)} - \mathbf{w} \right\rangle + \frac{\rho}{2} \left\| \mathbf{x}^{(k+1)} - \mathbf{w} \right\|^2. \end{cases}$$

All the solutions of these minimization problems have closed forms, namely

$$\begin{cases} \mathbf{x}^{(k+1)} = (A^T A + 2\rho I)^{-1} (A^T \mathbf{b}^\delta + \rho \mathbf{y}^{(k)} - \boldsymbol{\lambda}_1^{(k)} + \rho \mathbf{w}^{(k)} - \boldsymbol{\lambda}_3^{(k)}), \\ \mathbf{z}^{(k+1)} = S_{\mu/\rho} \left(L_\omega \mathbf{y}^{(k)} - \boldsymbol{\lambda}_2^{(k)} / \rho \right), \\ \mathbf{y}^{(k+1)} = (L_\omega^T L_\omega + I)^{-1} (L_\omega^T (\mathbf{z}^{(k+1)} + \boldsymbol{\lambda}_2^{(k)} / \rho) + \mathbf{x}^{(k+1)} + \boldsymbol{\lambda}_1^{(k)} / \rho), \\ \mathbf{w}^{(k+1)} = \left(\mathbf{x}^{(k+1)} + \boldsymbol{\lambda}_3 / \rho \right)_+, \\ \boldsymbol{\lambda}_1^{(k+1)} = \boldsymbol{\lambda}_1^{(k)} + \rho (\mathbf{x}^{(k+1)} - \mathbf{y}^{(k+1)}), \\ \boldsymbol{\lambda}_2^{(k+1)} = \boldsymbol{\lambda}_2^{(k)} + \rho (\mathbf{z}^{(k+1)} - L_\omega \mathbf{y}^{(k+1)}), \\ \boldsymbol{\lambda}_3^{(k+1)} = \boldsymbol{\lambda}_3^{(k)} + \rho (\mathbf{x}^{(k+1)} - \mathbf{w}^{(k+1)}), \end{cases}$$

where S_μ denotes the soft-thresholding operator with parameter μ, defined by

$$S_\mu(\mathbf{x}) = \text{sign}(\mathbf{x})(|\mathbf{x}| - \mu)_+,$$

and where the operations are meant element-wise and $(x)_+ = \max\{x, 0\}$ is the metric projection into the nonnegative cone. Note that each iteration requires the solution of two linear systems. The linear system involving A can be easily solved using the `fft` algorithm if periodic BCs are employed; see above. If other BCs are employed, then the structure of the matrix, in general, does not allow us to use fast transforms for the solution of the linear system. Nevertheless, this system can be solved by an iterative method using a circulant preconditioner. On the other hand, the solution of the linear system with the L_ω-matrix can be easily computed using the `lsqr` algorithm applied to the equivalent least-squares problem since the matrix L_ω is extremely sparse; see below and, e.g., [6] for a discussion.

We would like to briefly discuss the use of the `lsqr` method for the solution of the system

$$(L_\omega^T L_\omega + I)\mathbf{y}^{(k+1)} = L_\omega^T (\mathbf{z}^{(k+1)} + \boldsymbol{\lambda}_2^{(k)} / \rho) + \mathbf{x}^{(k+1)} + \boldsymbol{\lambda}_1^{(k)} / \rho. \tag{3.4}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The linear system of equations (3.4) is equivalent to the least-squares problem

$$\begin{aligned}\mathbf{y}^{(k+1)} &= \arg\min_{\mathbf{y}} \left\| \begin{bmatrix} L_\omega \\ I \end{bmatrix} \mathbf{y} - \begin{bmatrix} \mathbf{z}^{(k+1)} + \boldsymbol{\lambda}_2^{(k)}/\rho \\ \mathbf{x}^{(k+1)} + \boldsymbol{\lambda}_1^{(k)}/\rho \end{bmatrix} \right\|^2 \\ &= \arg\min_{\mathbf{y}} \left\| \widehat{L}\mathbf{y} - \widehat{\mathbf{v}}^{(k)} \right\|^2. \end{aligned} \tag{3.5}$$

The `lsqr` algorithm is an iterative method that determines an approximate solution of (3.5) in a Krylov subspace. In particular, at its jth iteration, the `lsqr` algorithm determines a solution of (3.5) in the Krylov subspace $\mathcal{K}_j(\widehat{L}^T\widehat{L}, \widehat{L}^T\widehat{\mathbf{v}}^{(k)})$, where

$$\mathcal{K}_j(\widehat{L}^T\widehat{L}, \widehat{L}^T\widehat{\mathbf{v}}^{(k)}) = \operatorname{span}\left\{ \widehat{L}^T\widehat{\mathbf{v}}^{(k)}, (\widehat{L}^T\widehat{L})\widehat{L}^T\widehat{\mathbf{v}}^{(k)}, \ldots, (\widehat{L}^T\widehat{L})^{j-1}\widehat{L}^T\widehat{\mathbf{v}}^{(k)} \right\}.$$

The `lsqr` method requires one matrix-vector product with $\widehat{L}$ and one with $\widehat{L}^T$ at each iteration. Therefore, since $\widehat{L}$ is extremely sparse, the flop count per iteration is $O(N)$. Moreover, `lsqr` is mathematically equivalent to the `cg` method applied to (3.4). However, its implementation is more stable. Nevertheless, the number of iterations required to converge is proportional to $\kappa(\widehat{L})$ which is small since, for N large enough, the spectrum of $\widehat{L}$ is included in the interval $[1, 5]$; see, e.g., [6] for a discussion on `lsqr` and `cg`.

We briefly discuss the bound for the spectrum of $\widehat{L} = I + L_\omega^T L_\omega$. Since the entries of Ω are positive, we have $\alpha_i := \sum_{j=1}^N \Omega_{i,j} = \sum_{j=1}^N |\Omega_{i,j}|$. The ith Geršgorin disk of L_ω is the interval

$$\left[-2\frac{\alpha_i}{\|\Omega\|_F}, 0 \right] \subseteq \left[-2\frac{\|\Omega\|_\infty}{\|\Omega\|_F}, 0 \right];$$

see, e.g., [48] for a discussion on Geršgorin disks. Hence, the spectrum of $I + L_\omega^T L_\omega$ is contained in $\left[1, 1 + 4\|\Omega\|_\infty^2/\|\Omega\|_F^2\right]$. Each entry in each row of Ω is smaller or equal to 1, and there are at most K non-vanishing entries per row, where K is a constant that depends exclusively on R and does not depend on N. Therefore, we have $\|\Omega\|_\infty \leq K$. On the other hand, assuming that the image is rescaled so that its maximum value is 1, we have that each nonzero entry of Ω is larger than $\mathrm{e}^{-1/\sigma}$. Observe that there are least k nonzero entries per row; therefore, we have $\|\Omega\|_F^2 \geq kN\mathrm{e}^{-2/\sigma}$. Similarly as above, the value of k depends only on R. Combining all the estimates above we obtain that

$$\frac{\|\Omega\|_\infty^2}{\|\Omega\|_F^2} \leq \frac{K^2}{kN\mathrm{e}^{-2/\sigma}} = O\left(\frac{1}{N}\right).$$

In particular, if N is large enough, then we have that $\|\Omega\|_\infty^2/\|\Omega\|_F^2 < 1$. Therefore, the spectrum of $I + L_\omega^T L_\omega$ is contained in $[1, 5]$. Note that this estimate is very rough, and in all the computed examples we have observed that the spectrum of $\widehat{L}$ is contained in the interval $[1, 5]$ for small values of N. We summarize our approach in Algorithm 2.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 2 Graph Laplacian image deblurring.

Input: $A \in \mathbb{R}^{N\times N}$, $\mathbf{b}^\delta \in \mathbb{R}^N$, $R>0$, $\sigma>0$, $\rho>0$, $\tau>0$, $K>0$, $\mu>0$
Output: $\mathbf{x} \in \mathbb{R}^{n\times n}$
Run Algorithm 1 to compute L_ω
$\mathbf{y}^{(0)} = \mathbf{w}^{(0)} = \boldsymbol{\lambda}_1^{(0)} = \boldsymbol{\lambda}_2^{(0)} = \boldsymbol{\lambda}_3^{(0)} = \mathbf{0}$
for $k = 0, \dots, K$
  $\mathbf{x}^{(k+1)} = (A^TA + 2\rho I)^{-1}(A^T\mathbf{b}^\delta + \rho\mathbf{y}^{(k)} - \boldsymbol{\lambda}_1^{(k)} + \rho\mathbf{w}^{(k)} - \boldsymbol{\lambda}_3^{(k)})$
  $\mathbf{z}^{(k+1)} = S_{\mu/\rho}\left(L_\omega\mathbf{y}^{(k)} - \boldsymbol{\lambda}_2^{(k)}/\rho\right)$
  $\mathbf{y}^{(k+1)} = (L_\omega^TL_\omega + I)^{-1}(L_\omega^T(\mathbf{z}^{(k+1)} + \boldsymbol{\lambda}_2^{(k)}/\rho) + \mathbf{x}^{(k+1)} + \boldsymbol{\lambda}_1^{(k)}/\rho)$
  $\mathbf{w}^{(k+1)} = \left(\mathbf{x}^{(k+1)} + \boldsymbol{\lambda}_3/\rho\right)_+$
  $\boldsymbol{\lambda}_1^{(k+1)} = \boldsymbol{\lambda}_1^{(k)} + \rho(\mathbf{x}^{(k+1)} - \mathbf{y}^{(k+1)})$
  $\boldsymbol{\lambda}_2^{(k+1)} = \boldsymbol{\lambda}_2^{(k)} + \rho(\mathbf{z}^{(k+1)} - L_\omega\mathbf{y}^{(k+1)})$
  $\boldsymbol{\lambda}_3^{(k+1)} = \boldsymbol{\lambda}_3^{(k)} + \rho(\mathbf{x}^{(k+1)} - \mathbf{w}^{(k+1)})$
  if $k>1$ & $\left\|\mathbf{x}^{k+1} - \mathbf{x}^k\right\| \le \tau\left\|\mathbf{x}^k\right\|$
    Exit
  end if
end for
$\mathbf{x} = \mathbf{x}^{(k+1)}$

Note that, since the functional minimized in (1.4) is convex, we can apply the following classical result for ADMM:

THEOREM 3.1 (see, e.g., Section 3.2 of Boyd et al. [7]). *With the notation of Algorithm 2, it holds that*

(i) $\lim_{k\to\infty}\left\|\begin{bmatrix} I & O \\ O & I \\ I & O\end{bmatrix}\begin{bmatrix}\mathbf{x}^{(k)}\\ \mathbf{z}^{(k)}\end{bmatrix} - \begin{bmatrix} I & O \\ L_\omega & O \\ O & I\end{bmatrix}\begin{bmatrix}\mathbf{y}^{(k)}\\ \mathbf{w}^{(k)}\end{bmatrix}\right\| = 0$, *i.e., the iterates approach feasibility as* $k\to\infty$;

(ii) $\lim_{k\to\infty}\frac{1}{2}\left\|A\mathbf{x}^{(k)} - \mathbf{b}^\delta\right\|^2 + \mu\left\|\mathbf{z}^{(k)}\right\|_1 + \iota_0(\mathbf{w}^{(k)}) = p^*$, *where* p^* *is the solution of the minimization problem (3.1);*

(iii) $\lim_{k\to\infty}\boldsymbol{\lambda}_k = \boldsymbol{\lambda}^*$, *where* $\boldsymbol{\lambda}^*$ *is a dual optimal point, i.e., a saddle point of* L_0.

We now briefly comment on this result. Point (i) shows that in the limit, the constraints on the variables in (3.2) are satisfied. Points (ii) and (iii) imply that the iterates converge to a solution of the problem. In particular, $\left\|A\mathbf{x}^{(k)} - \mathbf{b}^\delta\right\|^2 + \mu\left\|\mathbf{z}^{(k)}\right\|_1 + \iota_0(\mathbf{w}^{(k)})$ converges to its minimum value as $k\to\infty$.

REMARK 3.2. ADMM can be slow to converge in certain scenarios. It is outside the scope of this paper to propose a fast algorithm for the solution of (3.1). Rather we wish to show the potential of L_ω as a regularization operator. Nevertheless, it is possible to accelerate the convergence of ADMM by extrapolation methods to improve the convergence rate of ADMM; see, e.g., [9, 30]. We would like to stress that, in our experiments, the proposed algorithm converges in a reasonable number of iterations, and, while it could benefit from an acceleration, accelerating it is not essential for our purposes.

4. Numerical examples. We now report some selected numerical examples to show the performances of the proposed method. We are particularly interested in illustrating that the graph Laplacian constructed in Algorithm 2 provides better reconstructions than the classical TV approach.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 4.1
Setting of the parameters in Algorithm 2.

Paramter	Value	Description
R	10	Support of the weight function in the Graph
σ	10^{-2}	Variance of the weight function in the Graph
ρ	10^{-1}	Augmentation parameter in ADMM
τ	10^{-4}	Stopping criterion for ADMM
K	3000	Maximum number of iterations
μ	Hand-tuned	Regularization parameter in (1.4)

We compare the results obtained using $L = L_{\mathrm{TV}}$ and $L = L_\omega$ in (1.4) with the solution $\mathbf{x}^*$ computed in line 9 of Algorithm 1. To compute the solution of (1.4) with $L = L_{\mathrm{TV}}$, we use the algorithm described in [22]. Moreover, we want to show the full potential of the proposed approach. To this aim, we construct the operator L_ω using the exact image $\mathbf{x}_{\mathrm{true}}$, and we denote it by $\widetilde{L}_\omega$. Obviously, this approach is not feasible in realistic scenarios, nevertheless, it allows us to exhibit the capabilities of the proposed method. In all our experiments we set the parameters as specified in Table 4.1.

Moreover, we wish to compare the proposed approach with other methods from the literature. We consider the following methods:

- Approximated Iterated Tikhonov with General Penalty term (APIT-GP) [8];
- CGLS, stopped with the discrepancy principle, from the IRtools toolbox [28];
- Constrained Projected Linearized Bregman [13].

The first two methods do not require an estimate of any parameter, while the last one requires the user to set the value of a regularization parameter. For the third method, we select the one that minimizes the RRE defined below.

We compare the considered methods in terms of accuracy using the Relative Restoration Error (RRE) computed as

$$\mathrm{RRE}(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{x}_{\mathrm{true}}\|_2}{\|\mathbf{x}_{\mathrm{true}}\|_2}$$

and the Peak Signal to Noise Ration (PSNR) defined by

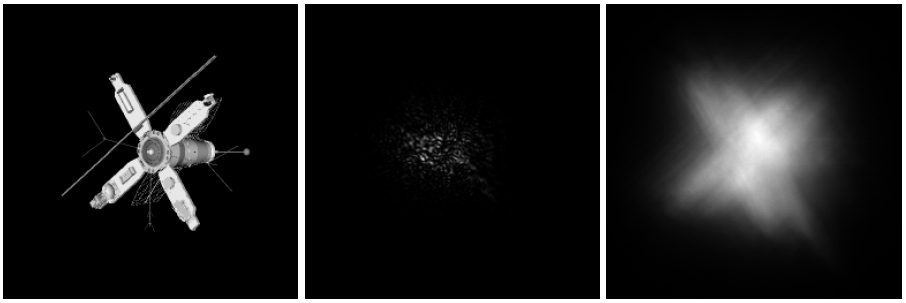
$$\mathrm{PSNR}(\mathbf{x}) = 20 \log_{10} \left(\frac{Nm}{\|\mathbf{x} - \mathbf{x}_{\mathrm{true}}\|_2} \right),$$

where m denotes the maximum value achievable by $\mathbf{x}_{\mathrm{true}}$. Moreover, we consider the Structure SIMilarity index (SSIM) constructed in [49]. The definition of the SSIM is extremely involved; here we simply recall that this statistical index measures how structurally similar two images are, in particular, the higher the SSIM the more similar the images are, and its highest achievable value is 1.

REMARK 4.1. A possible approach to further improve the quality of the computed restoration is to use the output of Algorithm 2 to construct a new graph Laplacian and run Algorithm 2 again with the "improved" regularization operator. In our experience, this, however, does not substantially improve the quality of the computed solution, and the additional computational cost is not justifiable. Moreover, this approach would require the estimate of the value of the regularization parameter μ twice (once per each run of Algorithm 2). Therefore, we do not present this algorithm here.

Example 1. Our first example is the `atmosphericBlur50` test case of the RestoreTools toolbox [3]. We report the exact image, the PSF, and the blurred and noisy image in

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) (b) (c)

FIG. 4.1. *Example 1: (a) True image (*256×256 *pixels). (b) PSF (*256×256 *pixels). (c) Blurred and noisy image (*256×256 *pixels with* $\delta \approx 0.01 \|\mathbf{b}\|_2$*).*

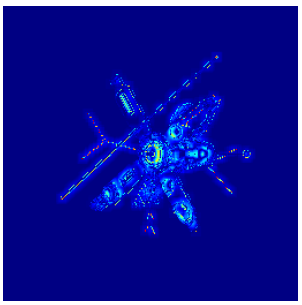
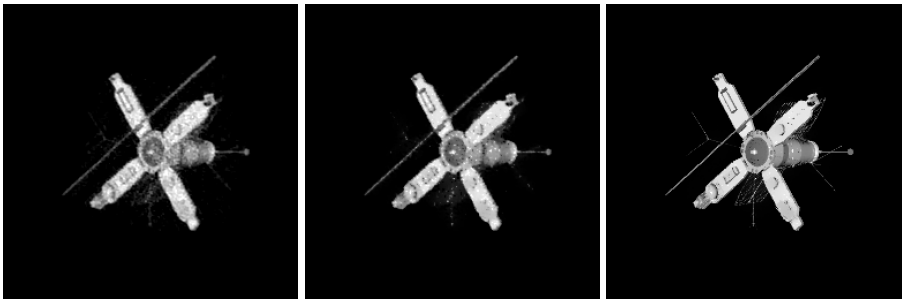


FIG. 4.2. *Example 1: Visualization of* $\left|\widetilde{L}_\omega \mathbf{x}^\dagger\right|$ *in the* `jet` *colormap. The color blue represents the* 0*s in the image.*



(a) (b) (c)

FIG. 4.3. *Example 1: Reconstructions: (a)* ℓ^2-ℓ^1 *with* $L = L_{\mathrm{TV}}$*. (b)* ℓ^2-ℓ^1 *with* $L = L_\omega$*. (c)* ℓ^2-ℓ^1 *with* $L = \widetilde{L}_\omega$*.*

Figure 4.1. The norm of the noise, denoted by δ, that corrupts the data is approximately 1% of the norm of the exact right-hand side $\mathbf{b}^\delta$.

To validate our model we display in Figure 4.2 a visualization of $\left|\widetilde{L}_\omega \mathbf{x}^\dagger\right|$. We can observe that it is extremely sparse, and only a few entries have a large modulus. Therefore, we expect our model to provide accurate reconstructions.

We report the obtained results with the considered methods in Table 4.2. We can observe that ℓ^2-ℓ^1 methods provide much more accurate results than the classical Tikhonov method, especially in terms of SSIM. The reconstruction obtained with $L = \widetilde{L}_\omega$, i.e., using the graph related to the exact image, is extremely accurate. However, this approach is not feasible in real

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 4.2
Comparison of the RRE, PSNR, and SSIM.

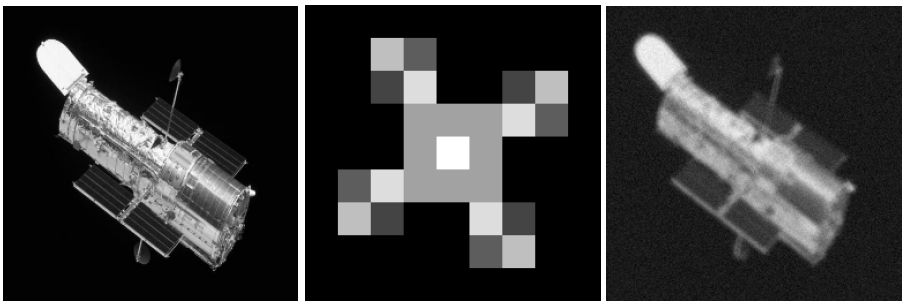
Example	Method	RRE	PSNR	SSIM
Example 1	Tikhonov	0.22299	26.663	0.55512
	ℓ^2-ℓ^1 with $L = L_{\text{TV}}$	0.19152	27.984	0.92623
	ℓ^2-ℓ^1 with $L = L_\omega$	0.17763	28.638	0.93971
	ℓ^2-ℓ^1 with $L = \widetilde{L}_\omega$	0.083333	35.212	0.98129
	APIT-GP	0.23292	26.284	0.89742
	CGLS	0.29454	24.246	0.38733
	Constrained Bregman	0.31758	23.591	0.85887
Example 2	Tikhonov	0.17352	25.735	0.55241
	ℓ^2-ℓ^1 with $L = L_{\text{TV}}$	0.15492	26.720	0.80458
	ℓ^2-ℓ^1 with $L = L_\omega$	0.14968	27.019	0.81256
	ℓ^2-ℓ^1 with $L = \widetilde{L}_\omega$	0.10096	30.439	0.89943
	APIT-GP	0.18024	25.405	0.75959
	CGLS	0.18487	25.185	0.50769
	Constrained Bregman	0.17074	25.875	0.77753
Example 3	Tikhonov	0.080283	33.715	0.72254
	ℓ^2-ℓ^1 with $L = L_{\text{TV}}$	0.068917	35.041	0.94873
	ℓ^2-ℓ^1 with $L = L_\omega$	0.060094	36.231	0.94809
	ℓ^2-ℓ^1 with $L = \widetilde{L}_\omega$	0.040382	39.684	0.96489
	APIT-GP	0.069238	35.001	0.96062
	CGLS	0.094625	32.288	0.73798
	Constrained Bregman	0.079316	33.821	0.94204
Example 4	Tikhonov	0.16236	27.160	0.73224
	ℓ^2-ℓ^1 with $L = L_{\text{TV}}$	0.15299	27.686	0.86899
	ℓ^2-ℓ^1 with $L = L_\omega$	0.14716	28.024	0.85765
	ℓ^2-ℓ^1 with $L = \widetilde{L}_\omega$	0.085936	32.695	0.93887
	APIT-GP	0.41090	19.104	0.62979
	CGLS	0.18810	25.891	0.72145
	Constrained Bregman	0.17140	26.700	0.81379

scenarios. Nevertheless, we can also observe that using $L = L_\omega$ improves the quality of the restoration with respect to the classic TV. This is confirmed by the visual inspection of the reconstructions in Figure 4.3. Comparing the reconstructions obtained by the ℓ^2-ℓ^1 methods, we can observe that the choice $L = L_{\text{TV}}$ leads to more noisy reconstructions than the ones obtained with the graph Laplacian. Moreover, we can observe that the ℓ^2-ℓ^1 methods are able to outperform all the considered benchmark algorithms both in terms of RRE and SSIM.

Example 2. For our second example, we consider the Hubble image in Figure 4.4(a), and we blur it with the PSF in Figure 4.4(b). We then add white Gaussian noise such that $\delta = 0.1 \|\mathbf{b}\|_2$ obtaining the blurred and noisy image in Figure 4.4(c).

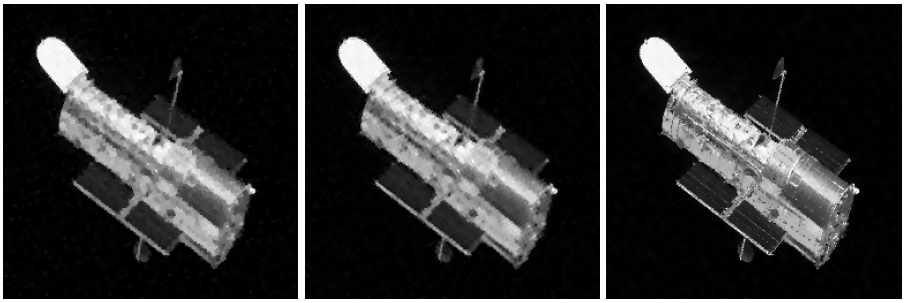
We compute approximate solutions with the considered algorithms and report the obtained RRE, PSNR, and SSIM in Table 4.2. We can observe that our proposal provides the best reconstruction both in terms of RRE (and therefore of PSNR) and SSIM. This is confirmed by the visual inspection of the reconstructions in Figure 4.5. We would like to stress that, similarly to the previous example, the unconstrained Tikhonov method computes extremely noisy reconstructions and that the ℓ^2-ℓ^1 methods outperform the benchmark algorithms.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) (b) (c)

FIG. 4.4. *Example 2: (a) True image (*256×256 *pixels). (b) PSF (*9×9 *pixels). (c) Blurred and noisy image (*256×256 *pixels with* $\delta = 0.1 \|\mathbf{b}\|_2$*).*



(a) (b) (c)

FIG. 4.5. *Example 2: Reconstructions: (a)* ℓ^2-ℓ^1 *with* $L = L_{\mathrm{TV}}$*. (b)* ℓ^2-ℓ^1 *with* $L = L_\omega$*. (c)* ℓ^2-ℓ^1 *with* $L = \widetilde{L}_\omega$*.*

Example 3. For our third example, we consider the Saturn image in Figure 4.6(a). We blur it with a non-symmetric PSF (see Figure 4.6(b)) and add 5% of white Gaussian noise, i.e., $\delta = 0.05 \|\mathbf{b}\|_2$ obtaining the image in Figure 4.6(c).

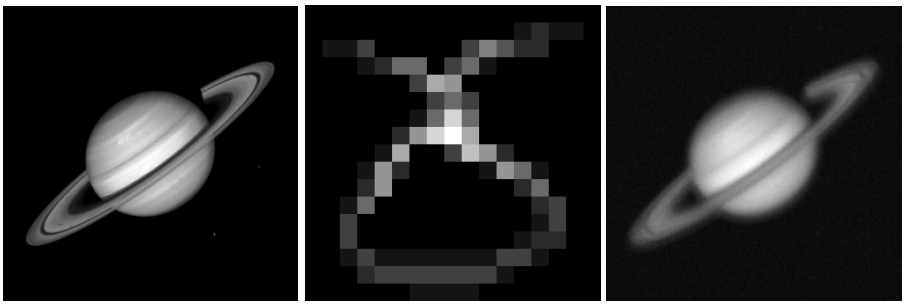
We report in Table 4.2 the obtained results with the considered algorithms. We observe that our proposal provides a very accurate reconstruction in terms of RRE and PSNR. However, the SSIM of the computed solution is slightly lower than the one obtained with the standard TV regularization. In Figure 4.7 we report all the computed solutions. Note that the reconstruction obtained with the classical TV regularization is affected by a very heavy stair-case effect that is avoided by our approach. This is evident in Figure 4.8, where we show blow-ups of the central part of the image of the exact solution and of the reconstructions obtained by TV regularization and our approach.

In this case we can observe that, similarly as above, the ℓ^2-ℓ^1 methods provide more accurate reconstructions than the ones computed by the benchmark methods. However, the SSIM obtained with APIT-GP is the highest among all tested algorithms. However, the reconstruction computed by APIT-GP is overly smoothed, and this may not be desirable in many situations.

Example 4. We consider the exact image in Figure 4.9(a), we blur it with an average PSF (see Figure 4.9(b)) and add white Gaussian noise so that $\left\|\mathbf{b} - \mathbf{b}^\delta\right\| = 0.03 \|\mathbf{b}\|$ obtaining Figure 4.9(c).

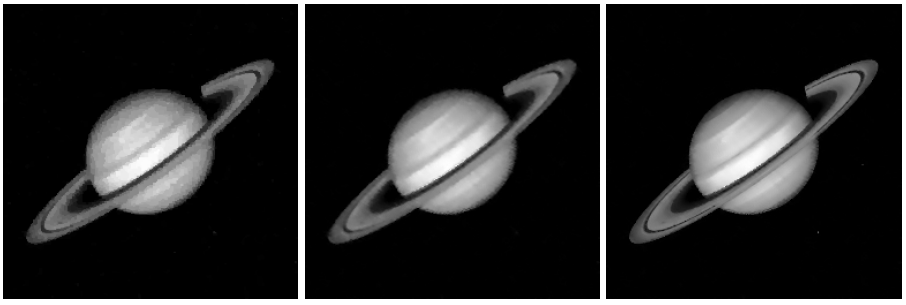
In Table 4.2 we note that the proposed algorithm with the perfect choice of $\widetilde{L}_\omega$ largely outperforms the other approaches furnishing a very accurate reconstruction of the proposed image. Moreover, the choice of $L = L_\omega$, which we recall does not require any a priori

ETNA
Kent State University and
Johann Radon Institute (RICAM)



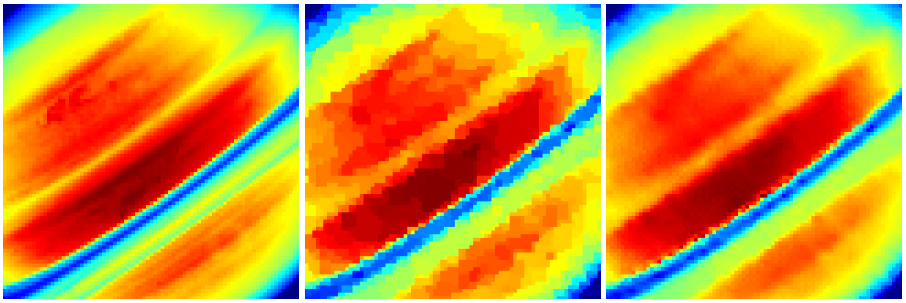
(a) (b) (c)

FIG. 4.6. *Example 3: (a) True image* (256×256 *pixels). (b) PSF* (17×17 *pixels). (c) Blurred and noisy image* (256×256 *pixels with* $\delta = 0.05\,\|\mathbf{b}\|_2$).



(a) (b) (c)

FIG. 4.7. *Example 3: Reconstructions: (a)* ℓ^2-ℓ^1 *with* $L = L_{\mathrm{TV}}$. *(b)* ℓ^2-ℓ^1 *with* $L = L_\omega$. *(c)* ℓ^2-ℓ^1 *with* $L = \widetilde{L}_\omega$.



(a) (b) (c)

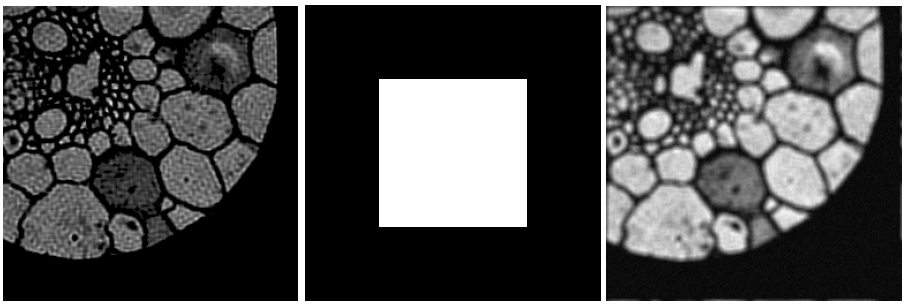
FIG. 4.8. *Example 3: Blow ups of the exact solution and of two reconstructions in the* `jet` *colormap: (a) True solution. (b)* ℓ^2-ℓ^1 *with* $L = L_{\mathrm{TV}}$. *(c)* ℓ^2-ℓ^1 *with* $L = L_\omega$.

information on the exact solution, is still more accurate than the classical TV. This is confirmed by the visual inspection of the reconstructions in Figure 4.10.

Finally, we observe that the APIT-GP method fails to provide an accurate reconstruction. In particular, the method fails to converge, reaching the maximum number of iterations.

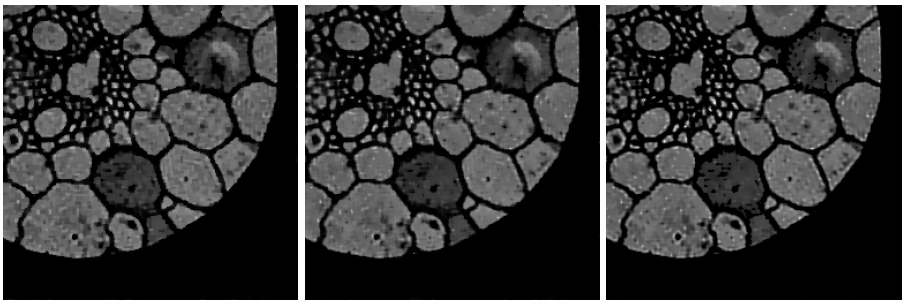
5. Conclusions. In this paper we have proposed a new regularization operator for ℓ^2-ℓ^1 minimization of image deblurring problems. The construction of this operator is automatic and extremely cheap to perform. We have shown that the proposed method outperforms the

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) (b) (c)

FIG. 4.9. *Example 4: (a) True image* (256×256 *pixels). (b) PSF* (12×12 *pixels). (c) Blurred and noisy image* (256×256 *pixels with* $\delta = 0.03\,\|\mathbf{b}\|_2$).



(a) (b) (c)

FIG. 4.10. *Example 4: Reconstructions: (a)* ℓ^2-ℓ^1 *with* $L = L_{\mathrm{TV}}$. *(b)* ℓ^2-ℓ^1 *with* $L = L_\omega$. *(c)* ℓ^2-ℓ^1 *with* $L = \widetilde{L}_\omega$.

classical TV approach. Matters of future research include the application of the proposed method to more general inverse problems as well as the integration of the considered method with the ℓ^p-ℓ^q minimization proposed in [12, 15, 34, 38] or with iterative regularization methods like, e.g., linearized Bregman splitting [11, 13, 17, 18, 19] and iterated Tikhonov regularization with general penalty terms [4, 5, 8, 10]. Another line of future research is the construction of more sophisticated graphs ω which can better exploit the structure of the given image itself. Such constructions may stem from a PDEs approach; see, e.g., [1, 29, 40].

Acknowledgment. The authors would like to thank the anonymous referees whose comments helped to improve the quality of this work.

REFERENCES

[1] A. ADRIANI, D. BIANCHI, AND S. SERRA-CAPIZZANO, *Asymptotic spectra of large (grid) graphs with a uniform local structure (Part I): Theory*, Milan J. Math., 88 (2020), pp. 409–454.
[2] Z.-Z. BAI, A. BUCCINI, K. HAYAMI, L. REICHEL, J.-F. YIN, AND N. ZHENG, *Modulus-based iterative methods for constrained Tikhonov regularization*, J. Comput. Appl. Math, 319 (2017), pp. 1–13.
[3] S. BERISHA AND J. G. NAGY, *Iterative methods for image restoration*, Tech. Rep., Department of Mathematics and Computer Science, Emory University, Atlanta, 2012. http://www.mathcs.emory.edu/~nagy/RestoreTools/IR.pdf.
[4] D. BIANCHI, A. BUCCINI, M. DONATELLI, AND S. SERRA-CAPIZZANO, *Iterated fractional Tikhonov regularization*, Inverse Problems, 31 (2015), Art. 055005, 34 pages.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

[5] D. BIANCHI AND M. DONATELLI, *On generalized iterated Tikhonov regularization with operator-dependent seminorms*, Electron. Trans. Numer. Anal., 47 (2017), pp. 73–99. http://etna.ricam.oeaw.ac.at/vol.47.2017/pp73-99.dir/pp73-99.pdf
[6] Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[7] S. BOYD, N. PARIKH, E. CHU, B. PELEATO, AND J. ECKSTEIN, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., 3 (2011), pp. 1–122.
[8] A. BUCCINI, *Regularizing preconditioners by non-stationary iterated Tikhonov with general penalty term*, Appl. Numer. Math., 116 (2017), pp. 64–81.
[9] A. BUCCINI, P. DELL'ACQUA, AND M. DONATELLI, *A general framework for admm acceleration*, Numer. Algorithms, 85 (2020), pp. 829–848.
[10] A. BUCCINI, M. DONATELLI, AND L. REICHEL, *Iterated Tikhonov with general penalty term*, Numer. Linear Algebra Appl., 24 (2017), Art. e2089, 12 pages.
[11] A. BUCCINI, Y. PARK, AND L. REICHEL, *Numerical aspects of the nonstationary modified linearized Bregman algorithm*, Appl. Numer. Math., 337 (2018), pp. 386–398.
[12] A. BUCCINI, M. PASHA, AND L. REICHEL, *Modulus-based iterative methods for constrained ℓ^p-ℓ^q minimization*, Inverse Problems, 36 (2020), Art. 084001, 26 pages.
[13] ———, *Linearized Krylov subspace Bregman iteration with nonnegativity constraint*, Numer. Algorithms, 87 (2021), pp. 1177–1200.
[14] A. BUCCINI AND L. REICHEL, *An ℓ^2-ℓ^q regularization method for large discrete ill-posed problems*, J. Sci. Comput., 78 (2019), pp. 1526–1549.
[15] ———, *An ℓ^p-ℓ^q minimization method with cross-validation for the restoration of impulse noise contaminated images*, J. Comput. Appl. Math., 375 (2020), Art. 112824, 16 pages.
[16] ———, *Generalized cross validation for ℓ^p-ℓ^q minimization*, Numer. Algorithms, 88 (2021), pp. 1595–1616.
[17] J.-F. CAI, S. OSHER, AND Z. SHEN, *Linearized Bregman iterations for frame-based image deblurring*, SIAM J. Imaging Sci., 2 (2009), pp. 226–252.
[18] ———, *Split Bregman methods and frame based image restoration*, Multiscale Model. Simul., 8 (2009), pp. 337–369.
[19] Y. CAI, M. DONATELLI, D. BIANCHI, AND T.-Z. HUANG, *Regularization preconditioners for frame-based image deblurring with reduced boundary artifacts*, SIAM J. Sci. Comput., 38 (2016), pp. B164–B189.
[20] L. CALATRONI, Y. VAN GENNIP, C. B. SCHÖNLIEB, H. M. ROWLAND, AND A. FLENNER, *Graph clustering, variational image segmentation methods and Hough transform scale detection for object measurement in images*, J. Math. Imaging Vision, 57 (2017), pp. 269–291.
[21] R. H. CHAN AND H.-X. LIANG, *Half-quadratic algorithm for ℓ_p-ℓ_q problems with applications to TV-ℓ_1 image restoration and compressive sensing*, in Efficient Algorithms for Global Optimization Methods in Computer Vision, A. Bruhn, T. Pock, and X.-C. Ta, eds., Lect. Notes Computer Sci. vol. 8293, Springer, Berlin, 2014, pp. 78–103.
[22] R. H. CHAN, M. TAO, AND X. YUAN, *Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers*, SIAM J. Imaging Sci., 6 (2013), pp. 680–697.
[23] M. DONATELLI AND L. REICHEL, *Square smoothing regularization matrices with accurate boundary conditions*, J. Comput. Appl. Math., 272 (2014), pp. 334–349.
[24] H. W. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of Inverse Problems*, Kluwer, Dordrecht, 1996.
[25] C. ESTATICO, S. GRATTON, F. LENTI, AND D. TITLEY-PELOQUIN, *A conjugate gradient like method for p-norm minimization in functional spaces*, Numer. Math., 137 (2017), pp. 895–922.
[26] C. FENU, L. REICHEL, AND G. RODRIGUEZ, *GCV for Tikhonov regularization via global Golub-Kahan decomposition*, Numer. Linear Algebra Appl., 23 (2016), pp. 467–484.
[27] C. FENU, L. REICHEL, G. RODRIGUEZ, AND H. SADOK, *GCV for Tikhonov regularization by partial SVD*, BIT, 57 (2017), pp. 1019–1039.
[28] S. GAZZOLA, P. C. HANSEN, AND J. G. NAGY, *IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems*, Numer. Algorithms, 81 (2019), pp. 773–811.
[29] G. GILBOA AND S. OSHER, *Nonlocal operators with applications to image processing*, Multiscale Model. Simul., 7 (2009), pp. 1005–1028.
[30] T. GOLDSTEIN, B. O'DONOGHUE, S. SETZER, AND R. BARANIUK, *Fast alternating direction optimization methods*, SIAM J. Imaging Sci., 7 (2014), pp. 1588–1623.
[31] G. H. GOLUB, M. HEATH, AND G. WAHBA, *Generalized cross-validation as a method for choosing a good ridge parameter*, Technometrics, 21 (1979), pp. 215–223.
[32] P. C. HANSEN, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, 1998.
[33] P. C. HANSEN, J. G. NAGY, AND D. P. O'LEARY, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

[34] G. HUANG, A. LANZA, S. MORIGI, L. REICHEL, AND F. SGALLARI, *Majorization-minimization generalized Krylov subspace methods for ℓ_p-ℓ_q optimization applied to image restoration*, BIT, 57 (2017), pp. 351–378.

[35] M. KELLER, D. LENZ, AND R. K. WOJCIECHOWSKI, *Graphs and Discrete Dirichlet spaces*, Springer, Cham, 2021.

[36] A. KHERADMAND AND P. MILANFAR, *Motion deblurring with graph Laplacian regularization*, in Digital Photography XI, N. Sampat, R. Tezaur, and D. Wüller, eds., SPIE Proceedings vol. 9404, SPIE, Bellingham, 2015, Art. 94040C, 8 pages.

[37] D. KRISHNAN AND R. FERGUS, *Fast image deconvolution using hyper-laplacian priors*, in Advances in Neural Information Processing Systems 22, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, eds., NIPS, La Jolla, 2009, pp. 1033–1041.

[38] A. LANZA, S. MORIGI, L. REICHEL, AND F. SGALLARI, *A generalized Krylov subspace method for ℓ_p-ℓ_q minimization*, SIAM J. Sci. Comput., 37 (2015), pp. S30–S50.

[39] F. LI AND M. K. NG, *Image colorization by using graph bi-Laplacian*, Adv. Comput. Math. 45 (2019), pp. 1521–1549.

[40] Y. LOU, X. ZHANG, S. OSHER, AND A. BERTOZZI, *Image recovery via nonlocal operators*, J. Sci. Comput., 42 (2010), pp. 185–197.

[41] F. G. MEYER AND X. SHEN, *Perturbation of the eigenvectors of the graph Laplacian: application to image denoising*, Appl. Comput. Harmon. Anal., 36 (2014), pp. 326–334.

[42] J. G. NAGY AND Z. STRAKOS, *Enforcing nonnegativity in image reconstruction algorithms*, in Mathematical Modeling, Estimation, and Imaging, D. C. Wilson, H. D. Tagare, F. L. Bookstein, F. J. Preteux, and E. R. Dougherty, eds., SPIE Proceedings vol. 4121, SPIE, Bellingham, 2000, pp. 182–190.

[43] J. PANG AND G. CHEUNG, *Graph Laplacian regularization for image denoising: analysis in the continuous domain*, IEEE Trans. Image Process., 26 (2017), pp. 1770–1785.

[44] G. PEYRÉ, S. BOUGLEUX, AND L. COHEN, *Non-local regularization of inverse problems*, in Computer Vision–ECCV 2008, D. Forsyth, P. Torr, and A. Zisserman, eds., Lect. Notes Computer Sci. vol. 5304, Springer, Berlin, 2008, pp. 57–68.

[45] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.

[46] A. SUSNJARA, N. PERRAUDIN, D. KRESSNER, AND P. VANDERGHEYNST, *Accelerated filtering on graphs using Lanczos method*, Preprint on arXiv, 2015. https://arxiv.org/abs/1509.04537

[47] D. TEWODROSE, *A survey on spectral embeddings and their application in data analysis*, Sém. Théor. Spectral et Géomét., 35 (2017-2019), pp. 197–244.

[48] R. S. VARGA, *Geršgorin and His Circles*, Springer, Berlin, 2014.

[49] Z. WANG, A. C. BOVIK, H. R. SHEIKH, AND E. P. SIMONCELLI, *Image quality assessment: from error visibility to structural similarity*, IEEE Trans. Image Process., 13 (2004), pp. 600–612.

[50] A. C. YAĞAN AND M. T. ÖZGEN, *A spectral graph Wiener filter in graph Fourier domain for improved image denoising*, in 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP), IEEE Conference Proceedings, IEEE, Los Alamitos, 2016, pp. 450–454.