Inverse Problems and Imaging
Volume 13, No. 5, 2019, 931–958
doi:10.3934/ipi.2019042

EDGE-ADAPTIVE ℓ_2 REGULARIZATION IMAGE RECONSTRUCTION FROM NON-UNIFORM FOURIER DATA

Victor Churchill*
Department of Mathematics
Dartmouth College
Hanover, NH 03755, USA

Rick Archibald
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37830, USA

Anne Gelb
Department of Mathematics
Dartmouth College
Hanover, NH 03755, USA

(Communicated by Weihong Guo)

Abstract. Signals and images recovered from edge-sparsity based reconstruction methods may not truely be sparse in the edge domain, and often result in poor quality reconstruction. Iteratively reweighted methods provide some improvement in accuracy, but at the cost of extended runtime. This paper examines such methods when data are acquired as non-uniform Fourier samples, and then presents a new non-iterative weighted regularization method that first pre-processes the data to determine the precise locations of the non-zero values in the edge domain. Our new method is both accurate and efficient, and outperforms reweighted regularization methods in several numerical experiments.

1. **Introduction.** Data for reconstruction of piecewise smooth functions and images are sometimes acquired as non-uniform Fourier samples. This is the case in non-Cartesian magnetic resonance imaging (MRI) and synthetic aperture radar (SAR). Since sparsity is inherent in the edge (jump discontinuity) domain of piecewise smooth functions and images, ℓ_1 norm based total variation (TV) regularization, [30], is commonly employed for reconstruction. The development of compressed sensing, [4, 5, 6, 11], has provided theoretical justification for using ℓ_1 regularization to promote sparsity in the appropiate domain. In some instances, however, the reconstructions produced via TV regularization based techniques are not as sparse in the edge domain as desired. This may be due to non-uniform sampling, noise, or the fact that the TV transform is not actually sparsifying with respect to functions that cannot be described as piecewise constant. Specifically, the TV

2010 *Mathematics Subject Classification.* Primary: 68U10, 65F22; Secondary: 42A10.
Key words and phrases. Image reconstruction, sparsity constraints, edge detection, iteratively reweighted ℓ_1 regularization.
Rick Archibald's work is sponsored by the Applied Mathematics Division of ASCR, DOE; in particular under the ACUMEN project (RA). Anne Gelb's work is supported in part by the grants NSF-DMS 1502640, NSF-DMS 1732434, AFOSR FA9550-18-1-0316 and AFOSR FA9550-15-1-0152.
* Corresponding author: victor.a.churchill.gr@dartmouth.edu.

transform does not transform all functions to the edge domain. What is likely a result of a combination of all these factors, the overall accuracy suffers. One popular approach for correcting this problem is to use an iterative reweighting scheme, [7, 9, 10, 23, 27, 35, 36]. These employ multiple runs of weighted ℓ_p minimization (typically $p = 1$ but $p = 2$ is also an option), where the weights typically help to locate the non-zero entries in the sparsity domain, and then apply regularization away from those locations in an iterative manner. Iterative reweighting methods have been shown to be more accurate than single-run TV methods, [7, 9], and have been applied to problems where data are acquired as uniform Fourier samples, [7]. The extension of these algorithms is straightforward for non-uniform Fourier data acquisition, although the implementation requires a non-uniform fast Fourier transform (NFFT), [13, 24, 26], and there are additional errors corresponding to the resulting fidelity term.[1] It is also important to note that in this investigation we are considering that the data acquired are noisy *continuous* non-uniform Fourier samples, which means that using the discrete NFFT generates additional model mismatch, [1].

This paper provides an alternative strategy to iterative reweighting. We propose an algorithm for signal and image reconstruction from non-uniform Fourier data that uses edge detection to indicate regions of sparsity in the TV domain and targets weighted ℓ_2 norm TV regularization appropriately. Unlike iterative reweighting, which requires multiple iterations of regularized minimization, there are only two steps to our method. The first step uses an edge detection technique to create a mask, i.e. a weighting matrix, which dictates where non-zero entries are expected in the TV domain. The edge map we employ in this paper is created from the Fourier data by solving an ℓ_1-regularized cost function. That said, we note that our method is compatible with any edge detection scheme and hence can easily be changed for a specific application. The second step uses this mask to target ℓ_2 regularization only in smooth regions, as specifically identified by the mask. In particular a smooth region is where the mask indicates zero response in the TV domain. Conversely, a non-zero TV response is indicated by the mask in edge regions. We clarify that these terms only specify what the mask indicates, rather than their true classification in the underlying image or signal. Put another way, our method uses regularization on targeted areas that are actually expected to be zero in the TV domain. Therefore, if the edge mask is indeed a good classifier of smooth and edge regions, it is appropriate to regularize using the ℓ_2 norm. Moreover, in edge regions, the method relies solely on the fidelity term since the mask returns zero weight value there. Finally we note the efficiency of our method as the reconstruction is performed by minimizing a single cost function regularized with the ℓ_2 (instead of the ℓ_1) norm. We call oour algorithm edge-adaptive ℓ_2 regularization.

There are several benefits to our proposed algorithm. First, it compares favorably in terms of accuracy (pointwise error) to iteratively reweighted methods. In particular, we test against a reweighted TV method similar to that in [7], which was used to reconstruct the Shepp-Logan phantom from discrete uniform Fourier coefficients. It also provides better resolution around jumps than iterative reweighting methods. Lastly from the accuracy perspective, this approach is particularly advantageous when the data contain additive noise, since the fidelity term can be weighted lightly against the regularization term, which strongly encourages noise reduction in smooth regions. As noted previously, our method is also more efficient

[1] This is the case whenever the acquired data are non-uniform Fourier samples, see e.g. [13].

to implement. Iterative reweighting methods require multiple iterations of solving regularized cost functions to identify the sparse regions. Our method needs only a single ℓ_1-regularized minimization for the pre-processing edge detection and a single ℓ_2-regularized minimization for the main reconstruction step. Further, we are able to use faster conjugate gradient descent optimization available for ℓ_2-regularized problems. Finally, there is a closed form solution to our problem, which may be valuable in some settings.

The rest of the paper is organized as follows: Section 2 covers the necessary background in image reconstruction from non-uniform Fourier data. Section 3 applies an iteratively reweighted ℓ_1 regularization method to this problem. Section 4 describes the edge-adaptive approach. Section 5 looks at numerical results. Conclusions and future work are in Section 6.

2. **Preliminaries.** In the one-dimensional case, we consider a piecewise smooth function $f : [-1, 1] \to \mathbb{R}$. Suppose we are given a finite sequence of non-uniform Fourier samples of f,

$$\hat{f}(\lambda_k) = \frac{1}{2}\int_{-1}^{1} f(x)e^{-\pi i\lambda_k x}dx, \tag{1}$$

where $\lambda_k \in \mathbb{R}$ and $k = -N, \ldots, N$. Specifically, we look at jittered sampling, defined by

$$\lambda_k = k + \frac{1 - 2\xi_k}{4}, \tag{2}$$

with $\xi_k \sim U([0, 1])$. We will also consider the case where the underlying Fourier data in (1) are noisy, given by

$$\hat{f}^{\eta}(\lambda_k) = \hat{f}(\lambda_k) + \eta_k, \tag{3}$$

for $k = -N, \cdots, N$. Here $\eta_k \sim \mathcal{CN}(0, \sigma^2)$, meaning η_k is a complex zero-mean Gaussian random variable with variance σ^2.

In two dimensions, we analogously consider a piecewise smooth function $f : [-1, 1]^2 \to \mathbb{R}$. Suppose we are given a finite sequence of non-uniform Fourier samples of f,

$$\hat{f}(\lambda_{\mathbf{k}}) = \frac{1}{4}\int_{-1}^{1}\int_{-1}^{1} f(x, y)e^{-\pi i\lambda_{k_1}x}e^{-\pi i\lambda_{k_2}y}dxdy, \tag{4}$$

where $\{\lambda_{\mathbf{k}} = (\lambda_{k_1}, \lambda_{k_2}) : k_1, k_2 = -N, \ldots, N\} \in \mathbb{R}^2$. The non-uniform jittered sampling pattern for $\lambda_{\mathbf{k}}$ is given by

$$\lambda_{\mathbf{k}} = \mathbf{k} + \frac{1 - 2\xi_{\mathbf{k}}}{4}, \tag{5}$$

with $\xi_{\mathbf{k}} \sim U([0, 1])^2$. The sampling patterns in (2) and (5), displayed in Figure 1, simulate Cartesian grid samples with slight deviations that sometimes occur in real world measurement systems.[2] We will also consider noisy two-dimensional Fourier data, $\hat{f}^{\eta}(\lambda_{\mathbf{k}})$, defined analogously to (3).

For ease of presentation, we begin by describing some known techniques for piecewise smooth function reconstruction in the one-dimensional case, where the

[2]More non-uniform sampling patterns, such as those that result from spiral MRI, are discussed in [17]. Our new technique is applicable in these cases, but for ease of presentation are not considered here.

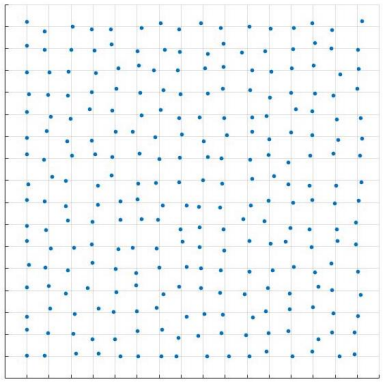

FIGURE 1. Non-uniform sampling $\lambda_{\mathbf{k}}$ as in (5).

acquired data are given in (1). These methods are easily extended to reconstruct two-dimensional images, which will be demonstrated in Section 3.

Let $\mathbf{f} = \{f(x_j) : j = -J, \dots, J\}$ and $\hat{\mathbf{f}} = \{\hat{f}(\lambda_k) : k = -N, \dots, N\}$. Since the underlying function f is piecewise smooth, it is sparse in the edge domain. That is, the number of jump discontinuities in the function is many fewer than $2J+1$. Hence ℓ_1 regularization provides an effective means for its reconstruction. In particular, f can be determined on a set of discrete grid points by solving the unconstrained optimization problem given by

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2^2 + \lambda ||\mathbf{T}\mathbf{g}||_1 \right). \tag{6}$$

Here $\mathcal{F}$ is the NFFT matrix (see e.g. [24, 26] for details about NFFT solvers), $\lambda > 0$ is the regularization parameter that balances fidelity, noise reduction, and sparsity, and $\mathbf{T}$ is a transformation to the edge domain. The choice of λ is typically problem dependent, [28]. We note that in this investigation we used single digit accuracy for the NFFT algorithm.

If f is a piecewise constant, for example a cross section of the Shepp Logan phantom seen in Figure 13, then the edge domain of f is precisely the TV domain of f. Hence, a standard choice for reconstruction is to solve the TV-regularized optimization problem

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2 + \lambda \sum_{j=-J}^{J-1} |\mathbf{g}_{j+1} - \mathbf{g}_j| \right), \tag{7}$$

which is frequently written as

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2 + \lambda ||\mathbf{D}\mathbf{g}||_1 \right). \tag{8}$$

Here $\mathbf{D}$ is simply the matrix that encodes the entry information from the sum in (7). Note that the sparsifying transformation in (8) is an approximation to the first derivative, penalizing high gradients in the function and therefore encouraging sparsity in the TV domain.

If f is a sparse signal, e.g. a spike train, then a standard reconstruction is the solution to the ℓ_1 regularized optimization problem

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2 + \lambda ||\mathbf{g}||_1 \right). \tag{9}$$

Finally, if f can be modeled as a piecewise polynomial, then a suitable regularization choice is high order total variation (HOTV), [1, 8], yielding

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2 + \lambda ||L^m\mathbf{g}||_1 \right). \tag{10}$$

Here L^m is the m^{th} order polynomial annihilation (PA) transform, [1, 2]. In general, L^m can be viewed as a normalized appproximation of the m^{th} derivative.[3] For example, when $m = 3$ we have

$$L^3 = \begin{bmatrix} -\frac{1}{2} & \frac{3}{2} & -\frac{3}{2} & \frac{1}{2} & 0 & \cdots & 0 \\ 0 & -\frac{1}{2} & \frac{3}{2} & -\frac{3}{2} & \frac{1}{2} & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & -\frac{1}{2} & \frac{3}{2} & -\frac{3}{2} & \frac{1}{2} \end{bmatrix}. \tag{11}$$

Using $L^0 = I$ gives (9), while using L^1 yields (7). Hence, going forward we will refer to a signal or image transformed by L^m as being in the gradient domain, and (10) as HOTV regularization.

Observe that the polynomial annihilation (PA) transform annihilates polynomials of degree $m-1$ in smooth regions. Thus using $m = 1$ is best for piecewise constant functions, $m = 2$ for piecewise linear functions, etc. In Section 5 we demonstrate that our method is robust to different choices of m. Finally we note that our method can be readily adapted for other sparsifying transformations.

For square two-dimensional images, we regularize in the x and y directions separately and solve

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2 + \lambda \left(||L^m\mathbf{g}||_1 + ||\mathbf{g}(L^m)^T||_1 \right) \right), \tag{12}$$

where $\mathbf{g}$ is $N \times N$, $\mathcal{F}$ is the 2D Fourier transform, and $\hat{\mathbf{f}}$ contains the collected 2D Fourier coefficients. Observe that norms in the 2D case are taken after vertical concatenation of the matrices. The regularization term $||L^m\mathbf{g}||_1$ penalizes gradients in the y direction and the regularization term $||\mathbf{g}(L^m)^T||_1$ penalizes approximate gradients in the x direction.[4] This is effectively an anisotropic (high order) TV formulation.

Remark 1. We choose the anisotropic rather than the isotropic formulation (see e.g. [30]) for several reasons. First, the separability of the directional penalties makes for a more cost effective minimization, which is not possible with isotropic TV. Second, the anisotropic PA transform formulation gives the possibility for high order derivative approximations, which better approximate the true edge domain in cases where the function to be reconstructed is not piecewise constant. Lastly,

[3] Indeed, although there are subtle differences in the derivations and normalizations, the PA transform can be thought of as a variant of HOTV. Because part of our investigation discusses parameter selection, which depends explicitly on $||L^m f||$, we will exclusively use the PA transform as it appears in [1] so as to avoid any confusion. Explicit formulations for the PA transform matrix can also be found in [1].

[4] The PA matrix L^m can be constructed for two dimensional images, [2]. However, in [1] it was demonstrated that splitting the dimensions was more cost effective and did not reduce the quality of the reconstruction.

while in unweighted schemes isotropic TV regularization is typically more accurate than anisotropic TV regularization, there is evidence that in the context of weighted TV regularization based on edges that the anisotropic is superior. For example, an iteratively reweighted method based on edge sparsity is proposed in [25], where both isotropic and anisotropic TV formulations are considered. The authors demonstrate that for edge-weighting methods, anisotropic TV is more accurate than the isotropic version. Hence we limit our investigation here to the anisotropic case simply to demonstrate the value of edge adaptive ℓ_2 regularization. In future investigations, we will also consider the isotropic formulation.

As noted previously, using (10) is effective in reconstructing piecewise smooth functions and images in a large number of applications. For a variety of reasons, however, the assumption that the function or image is sparse in the edge domain is often flawed. One reason is noise, which will immediately degrade the edge sparsity of the solution to (10). For TV regularization, the assumption that the transformed image is sparse is often inadequate due to smooth variation away from jump discontinuities. This is somewhat mitigated by HOTV regularization with $m > 1$. However, if due to lack of resolution the image has variation not accounted for away from discontinuities, even high order transformations will not produce the desired sparsity. Another source of error is non-uniform sampling, since some compromise in accuracy is necessary to maintain the efficiency of the NFFT. Finally, all ℓ_1 based methods suffer from the fact that the ℓ_1 norm penalizes large magnitudes more heavily, therefore affecting large jump discontinuities more than small ones. This is of particular concern when there are multiple scales in the function.

A popular approach to mitigating error from issues such as noise, lack of resolution, and magnitude dependence is to use a scheme that employs iteratively reweighted (IR) regularization, [7, 9, 10, 23, 27, 35, 36]. In these methods, multiple passes of TV or HOTV regularization with weighted ℓ_p norms are used to "narrow in" on large values in the gradient domain. The weight at each point on the spatial grid is typically inversely proportional to the magnitude of that point in the gradient domain of the previous iteration. That is, the regularization is more strongly enforced at points deemed by weighting as non-jumps and more weakly enforced at those identified by the weighting as jumps. In this way, iterative reweighting more democratically penalizes high gradients and regularizes based on the spatial distribution of the sparsity. These iterative methods are typically more accurate than single pass methods, [7, 9]. We will use the next section to discuss IR methods in more detail.

3. Iteratively reweighted regularization methods.

As explained in [4], reconstructing an image or function via solving

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2^2 + \lambda ||L^m \mathbf{g}||_0 \right), \tag{13}$$

where $||\cdot||_0$ counts non-zero values, promotes the most sparsity in the gradient domain of the image. However, this combinatorial problem is NP-hard. In (10), the ℓ_1 term acts as a convex surrogate for the ℓ_0 term, making the problem easier to solve. However, it does not encourage sparsity in the gradient domain as much. Naturally, this begs the question of whether there are better surrogates that generate solvable optimization problems.

The approach of [7] is to regularize using the log-sum function, a concave penalty function that more closely resembles the ℓ_0 norm and is therefore more sparsity-inducing. That reconstruction would be the solution to the optimization problem

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2^2 + \sum_{j=-J}^{J} \log(|(L^m\mathbf{g})_j| + \epsilon) \right), \tag{14}$$

where $\epsilon > 0$ is a small parameter to stay within the domain of the logarithm. Since the log-sum function is nonconvex, (14) is difficult solve. Instead, we can approximate it with a series of weighted ℓ_1 based minimizations of the form

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2^2 + \lambda||WL^m\mathbf{g}||_1 \right), \tag{15}$$

where W is a diagonal matrix of weights. The main idea is that large weights can be used to discourage non-zero entries in the gradient domain, while small weights can be used to encourage non-zero entries. Hence this method will penalize non-zero gradient magnitudes more fairly, removing the magnitude dependence of the unweighted ℓ_1 norm. To achieve this, weights inversely proportional to the gradient magnitudes of the previous iteration are used. In this way, IR methods encourage sparsity in the gradient domain by regularizing less heavily in regions where jumps are suspected and more heavily in regions where no jumps are suspected.

Algorithms 1 and 2 are modified versions of the iterative weighting method used in [7] for one- and two-dimensional functions, respectively. They are formulated using the PA transform L^m. When $m = 1$, we will refer to Algorithms 1 and 2 as reweighted total variation (RWTV) methods. For $m \geq 2$, they will be called reweighted high order total variation (RWHOTV) methods.

Algorithm 1 IR ℓ_1 regularization reconstruction in one dimension

1: Set $\ell = 0$ and $w_j^{(0)} = 1$ for $j = -J, \ldots, J$. Fix the regularization parameter $\rho > 0$, the weighting parameter $\epsilon > 0$, and an appropriate PA order m.

2: Solve the weighted regularization minimization problem

$$\mathbf{f}^{(\ell)} = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2^2 + \rho||W^{(\ell)}L^m\mathbf{g}||_1 \right) \tag{16}$$

where $W^{(\ell)} = \text{diag}(w^{(\ell)})$.

3: Update the weights. For each $j = -J, \ldots, J$,

$$w_j^{(\ell+1)} = \frac{1}{|(L^m\mathbf{f}^{(\ell)})_j| + \epsilon}. \tag{17}$$

4: Terminate on convergence or when ℓ attains a pre-specified maximum number of iterations ℓ_{max}. Otherwise, increment ℓ and go to step 2.

As explained in [7], the main advantages of this method are increased accuracy using the same number of Fourier coefficients and removal of magnitude dependence of the unweighted ℓ_1 norm. A chief example of when this method works very well is found in Section 3.6, in particular Figure 10, of [7]. These advantages are balanced with some disadvantages. The runtime is increased ℓ_{max} times for this method, since each iteration requires an ℓ_1 minimization step. In addition, this method introduces another parameter ϵ. No comprehensive method for choosing this parameter is provided in [7] and the success of this algorithm depends on an appropriate choice.

Algorithm 2 IR ℓ_1 regularization reconstruction in two dimensions

1: Set $\ell = 0$ and $v_{i,j}^{(0)} = 1$ for $i,j = -J \dots J$ and $w_{i,j}^{(0)} = 1$ for $i,j = -J \dots J$. Fix the regularization parameter $\rho > 0$, the weighting parameter $\epsilon > 0$, and an appropriate PA order m.

2: Solve the weighted regularization minimization problem

$$\mathbf{f}^{(\ell)} = \arg\min_{\mathbf{g}} \left\{ ||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2^2 + \rho \left(\sum_{i=-J}^{J} \sum_{j=-J}^{J} v_{i,j}^{(\ell)} |(L^m \mathbf{g})_{i,j}| + \sum_{i=-J}^{J} \sum_{j=-J}^{J} w_{i,j}^{(\ell)} |(\mathbf{g}(L^m)^T)_{i,j}| \right) \right\} \tag{18}$$

3: Update the weights. For each (i,j) such that $i,j = -J, \dots, J$,

$$v_{i,j}^{(\ell+1)} = \frac{1}{|(L^m \mathbf{f}^{(\ell)})_{i,j}| + \epsilon} \quad \text{and} \quad w_{i,j}^{(\ell+1)} = \frac{1}{|(\mathbf{f}^{(\ell)}(L^m)^T)_{i,j}| + \epsilon}. \tag{19}$$

4: Terminate on convergence or when ℓ attains a pre-specified maximum number of iterations ℓ_{max}. Otherwise, increment ℓ and go to step 2.

Finally, there still appear to be clear sources of error generated by this method, which we will now explore.

As prototype examples to test the IR methods, we consider

Example 1.

$$f_1(x) = \begin{cases} \cos(x/2) & x \geq 0 \\ -\cos(x/2) & x < 0 \end{cases}, \tag{20}$$

Example 2.

$$f_2(x) = \begin{cases} \frac{3}{2} & -\frac{3\pi}{4} \leq x < -\frac{\pi}{2} \\ \frac{7}{4} - \frac{x}{2} + \sin(7x - \frac{1}{4}) & -\frac{\pi}{4} \leq x < \frac{\pi}{8} \\ \frac{11x}{4} - 5 & \frac{3\pi}{8} \leq x < \frac{3\pi}{4} \\ 0 & else \end{cases}, \tag{21}$$

and

Example 3.

$$f_3(x,y) = \begin{cases} \cos(\pi(x^2+y^2)) & x^2+y^2 \leq \frac{1}{2} \\ \cos(\pi(x^2+y^2) - \frac{\pi}{2}) & x^2+y^2 > \frac{1}{2} \end{cases}. \tag{22}$$

One issue with Algorithms 1 and 2 is that when noise is present there are non-zero weights being applied in areas of smooth variation (and no variation), resulting in what amounts to false jump identifications and ultimately oscillation in the reconstruction. The oscillations caused by noise in the initial solution are propagated through all the iterations via the weighting matrix. The algorithm has no way to validate whether these oscillations are from actual smooth variation, a jump discontinuity, or noise. Figure 2 demonstrates the use of Algorithm 1 for $f_1(x)$ and $f_2(x)$ when the given Fourier data (1) is noise-free and when complex zero-mean Gaussian noise is added as in (3). Notice how the oscillations in the reconstruction increase where the function has more smooth variation.

The source of this error is from points weighted between 0 and $\frac{1}{\epsilon}$. These weights can indicate either a small jump discontinuity or smooth variation that is beyond

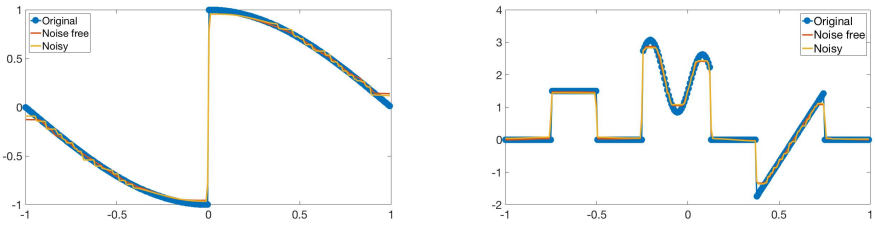

FIGURE 2. $f_1(x)$ (top) and $f_2(x)$ (bottom) reconstructed via Algorithm 1 using $m = 1$ from 257 Fourier modes on 257 grid points. The red reconstructions indicate recovery from noise-free data as in (1), while yellow reconstructions are recovered from Fourier coefficients with zero-mean complex Gaussian noise added as in (3). Here we use a signal to noise ratio (SNR) of 20 dB. For $f_1(x)$, we used parameters $\rho = 1$, $\ell_{max} = 25$, and $\epsilon = 1.9$. For $f_2(x)$, we used parameters $\rho = 1$, $\ell_{max} = 25$, and $\epsilon = 2.9$.

the resolution of the problem, which may or may not be attributable to noise. If there is a small jump, the iterative reweighting strategy still regularizes at that point, albeit relatively less. If it is caused by noise or more simply smooth variation in the function itself, then the algorithm regularizes less at that point for no reason. This will automatically reduce the algorithm's ability to separate the true scales of the underlying image by causing what amounts to false jump identifications, leading to an overall less accurate reconstruction.

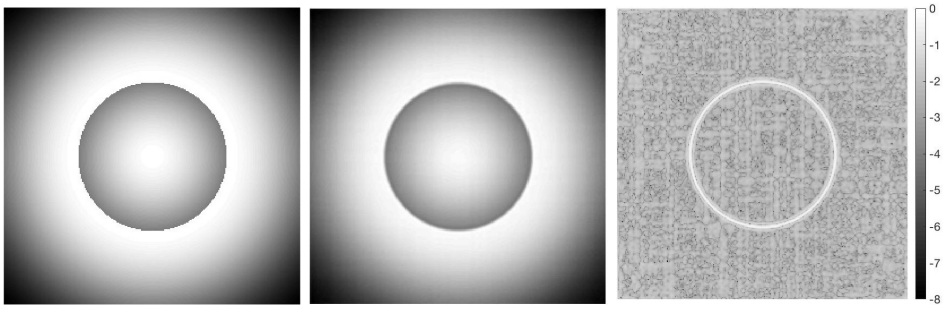

FIGURE 3. (Left) 257×257 pixel function $f_3(x, y)$; (Middle) reconstruction via Algorithm 2 with $m = 2$ from 257×257 jittered Fourier coefficients on 257×257 grid points; (Right) pointwise error plotted on a logarithmic scale. The algorithm parameters used are $\rho = .01$, $\epsilon = .9$, and $\ell_{max} = 5$.

A two-dimensional example of this phenomenon can be seen in Figure 3, which shows a reconstruction via Algorithm 2 of the 257×257 pixel function $f_3(x, y)$. Figure 4 elucidates the cause of these inaccuracies. The left image shows the ideal weights $w_{i,j}$ as in (19). By ideal, we mean computed using (19) directly from the exact function $f_3(x, y)$. The goal of Algorithm 2 is to converge to these weights, since that would mean the algorithm generated $\mathbf{f}^{(\ell)}$ very close to the true $f_3(x, y)$. On the right we display the actual weights $w_{i,j}$ computed by Algorithm 2. The

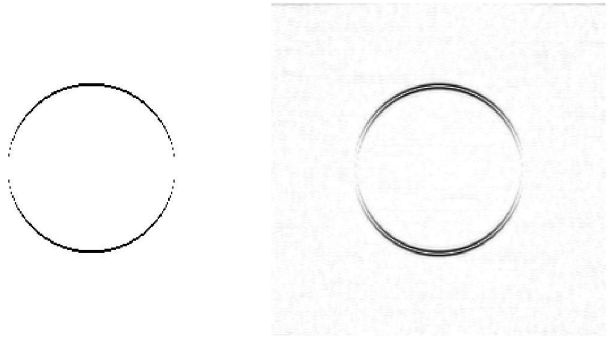

FIGURE 4. (Left) Ideal weighting matrix for the y-direction edges computed from (19) using the exact 257×257 two-dimensional function $f_3(x, y)$; (Right) final weighting matrix for the y-direction edges produced by Algorithm 2. The minimum weight of 0 is indicated by black while maximum weight of $\frac{1}{\epsilon} \approx 1.11$ is indicated by white, while gray indicates a weight in between 0 and $\frac{1}{\epsilon}$.

black or gray points in the right weighting matrix different from those obtained by the ideal left weighting matrix are effectively being identified by Algorithm 2 as either a jump or variation beyond the resolution of the problem. Since there are no jumps outside of the black area in this function, this weighting matrix shows that the algorithm is falsely identifying many jumps. In this context, that means that this algorithm is regularizing less in areas it shouldn't be, which leads to an overall less accurate reconstruction.

In what follows we demonstrate how Algorithms 1 and 2 can be improved upon in terms of accuracy, simplicity, efficiency and robustness.

4. **Edge-adaptive ℓ_2 regularization.** As discussed in [7], without prior information about the non-zero elements in a sparse signal or similarly the locations of edges in an image, it is effective to choose regularization weights iteratively. However, when starting with Fourier data the weighting scheme adopted by Algorithms 1 and 2 is likely not the most direct way to penalize non-zero locations in the sparsity domain. Therefore we take a more direct approach. Specifically we locate the edges directly from the Fourier data, as in [17], and then construct the regularization weights to be zero-valued anywhere an edge is detected and non-zero at all other points. Unlike iterative reweighting, which requires multiple ℓ_1 minimizations, there are only two minimization steps in our new method. First, we perform an ℓ_1 regularization based edge detection to determine where the support is in the sparsity domain. We note that while ℓ_1-regularized edge detection is a good choice for non-uniform Fourier data, our method is flexible and allows the edge map to be generated however the user prefers for their specific application. We then create a mask, i.e. a weighting matrix, based on the detected edges. This mask allows us to target ℓ_2 regularization only to smooth regions of the function in a second minimization step. Recall that here smooth regions are those which the mask indicates are zero in the sparsity domain. In this way, our method uses regularization on regions which the edge detection suggests have zero intensity. Therefore the usual compressed sensing arguments for using the ℓ_1 norm as a surrogate for the ℓ_0 norm

no longer apply. In particular, it is just as appropriate to use the ℓ_2 norm to minimize something that is supposed to be zero rather than sparse, and it is much more cost efficient than using the ℓ_1 norm. We note that the method relies solely on the fidelity term in the edge regions, which recall are the regions the mask indicates should be nonzero in the gradient domain. We note again that the edge and smooth region labels do not indicate regions with or without a true jump discontinuity, and are only as faithful to such a designation as the mask and the edge detection used to create it.

4.1. **Edge detection from non-uniform Fourier data.** The edge adaptive ℓ_2 regularization image reconstruction technique depends heavily on the selection of a weighting mask, which is explicitly determined by the edges recovered from the given non-uniform Fourier data. While there have been a number of algorithms designed to extract edges from Fourier data, we will use the concentration factor (CF) method introduced in [18] for uniformly sampled Fourier data and adapted in [17] for the non-uniform case. It is briefly described below.

Let us first consider a one-dimensional periodic piecewise smooth function $f : [-1, 1] \to \mathbb{R}$. We define the jump function, $[f]$, as

$$[f](x) = f(x^+) - f(x^-), \tag{23}$$

the difference between the left- and right-hand limits of the function. For x at which f is continuous, $[f](x) = 0$. At a discontinuity, $[f](x)$ is the signed magnitude of the jump. Suppose we are given $2J + 1$ grid points, $x_j = \frac{j}{J}$, $j = -J, \ldots, J$. Assuming that the discontinuities $\{\xi_l\}_{l=1}^L$ of f are separated such that there is at most a single jump per cell, $I_j = [x_j, x_{j+1})$, we can write

$$[f](x) = \sum_{j=-J}^{J-1} [f](x_j)\delta_{x_j}(x). \tag{24}$$

where the coefficients $[f](x_j)$ is the value of the jump that occurs within the cell I_j and $\delta_\xi(x)$ is the indicator function with $\delta_\xi(x) = 1$ when $x = \xi$ and 0 otherwise.

The concentration factor (CF) edge detection method, [15, 18, 19, 20], approximates (24) from the first $2N + 1$ uniform Fourier coefficients given in (1) where $\lambda_k = k$ as

$$S_N^\sigma[f](x) = i \sum_{k=-N}^{N} \hat{f}(k)\mathrm{sgn}(k)\sigma(k)e^{\pi i k x}. \tag{25}$$

Here $\sigma = \sigma(k)_{k=-N}^N$, coined the concentration factor in [18], satisfies certain admissibility conditions. The convergence of (25) depends on the particular choice of σ. The concentration factor σ is effectively a bandpass filter that concentrates at the singular support of the underlying function. Hence (25) is a filtered partial Fourier sum that approximates the jump function. However, there is a lot of flexibility in how σ may be chosen. We delay explaining this choice, though, as significantly the CF edge detection method cannot be extended directly to non-uniform Fourier coefficients because $\{e^{\pi i \lambda_k x}\}_{k=-N}^N$ is not an orthogonal basis. In the next subsection, we explain the modifications needed to apply CF edge detection to non-uniform Fourier data, as well as how to optimally design a concentration factor for this application.

4.1.1. *Concentration factors for non-uniform Fourier data.* For ease of presentation, let us first assume that there is only one discontinuity at $x = \xi \in (-1, 1)$. The extension to general piecewise functions is straightforward. The jump function in (24) is now simply

$$[f](x) = [f](\xi)\delta_\xi(x). \tag{26}$$

As described in [17], it is possible to project the non-uniform Fourier data onto $\{e^{im\pi x}\}_{m=-M}^{M}$ as

$$T_N f = \sum_{|m|<M} \sum_{|k|<N} \hat{f}(\lambda_k) b_{m,k} e^{i\pi m x}. \tag{27}$$

Here $b_{m,k}$ is the $(m,k)^{th}$ entry of the Moore-Penrose pseudo-invese of the matrix given by

$$\mathbf{\Psi} = [\langle e^{\pi i \lambda_k x}, e^{\pi i m x} \rangle],\ |m| \le M,\ |k| \le N.$$

Note that

$$\langle e^{\pi i \lambda_k x}, e^{\pi i m x} \rangle = \int_{-1}^{1} e^{\pi i (\lambda_k - m) x} = 2sinc(\lambda_k - m).$$

We remark that other projections are also possible, and may lead to faster and more robust approximations, see [16]. For our purposes the projection given by $T_N f$, which is simply a uniform resampling (URS) approximation (see e.g. [29]), is appropriate.[5] To approximate the jump function from non-uniform coefficients, we can use (27) and seek σ such that

$$T_N^\sigma f(x) = \sum_{|k|<N} \sigma(k) \hat{f}(\lambda_k) \tilde{\varphi}_{M,k}(x) \approx [f](\xi)\delta_\xi(x). \tag{28}$$

where $\tilde{\varphi}_{M,k}(x) = \sum_{|m|\le M} b_{m,k} e^{i\pi m x}$. Note that we have folded in $i\text{sgn}(k)$ into $\sigma(k)$.

Observe that if f is piecewise-analytic, it can be approximated by a scaled and shifted ramp function given by

$$f(x) \approx a r_\xi(x). \tag{29}$$

Here $r_\xi(x) = r(x - \xi)$ for $\xi \in (-1, 1)$ with

$$r(x) = \begin{cases} -\frac{x+1}{2} & \text{if } x \in [-1, 0] \\ -\frac{x-1}{2} & \text{if } x \in (0, -1], \end{cases} \tag{30}$$

and a corresponds to the jump value $[f](\xi)$, that is $[f](\xi)\delta_\xi(x) = a[r_\xi](x)$. Hence it is apparent that while (29) is only a first order approximation of $f(x)$, it is perfectly reasonable to compute $[f](x)$. Specifically, from (29) we have

$$[f](x) = [f](\xi)[r_\xi](x). \tag{31}$$

Using (31) in (28) we have

$$[f](\xi) T_N^\sigma r_\xi(x) = [f](\xi) \sum_{|k|\le N} \sigma(k) \widehat{r_\xi}(\lambda_k) \tilde{\varphi}_{M,k}(x) \approx [f](\xi)\delta_\xi(x), \tag{32}$$

where $\widehat{r_\xi}$ are the Fourier coefficients of $r_\xi(x)$. Translating this system yields

$$\sum_{|k|\le N} \sigma(k) \hat{r}(\lambda_k) \tilde{\varphi}_{M,k}(x) \approx \delta_0(x), \tag{33}$$

[5] As noted in [17], to ensure that Ψ is not ill-conditioned, one must choose M to reflect the amount of non-uniformity in the original sampling. Since only jittered sampling is discussed in this investigation, choosing $M = N$ is reasonable.

where $\hat{r}(\lambda_k)$ are the Fourier coefficients of $r(x)$ explicitly given by

$$\hat{r}(\lambda_k) = \begin{cases} 0 & \lambda_k = 0 \\ \frac{(\sin(\pi\lambda_k) - \pi\lambda_k)i}{(\pi\lambda_k)^2} & \text{otherwise.} \end{cases} \tag{34}$$

As was demonstrated in [17], one possibility to obtain the concentration factor vector σ is to solve (33) as a linear system on the set of grid points x_j, $j = -J, \cdots, J$. This may be preferred when λ_k is sampled sparsely (i.e. *not* a jittered sampling pattern), but as is demonstrated below, an explicit solution for the concentration factor σ can also be directly obtained from (33).

We note that it is difficult to approximate $\delta_0(x)$ from (33) since it only has non-trivial values on a set of measure zero and therefore does not have a non-trivial Fourier expansion. Hence we first regularize the problem by introducing a smooth function $h_0(x) \approx \delta_0(x)$. For example, in our experiments we employ

$$h_0(x) = \frac{1}{2N+1}\left(2\frac{\sin(N\pi x)}{\pi x} - 1\right), \tag{35}$$

which has some beneficial cancellation properties in the uniform case ($\lambda_k = k$) when the SNR is high, [19]. On the other hand, using a Gaussian function

$$h_0(x) = \exp\left(-\alpha\left(\frac{x}{\beta}\right)^2\right)$$

reduces the effects of noise but does not resolve the discontinuities as well. In [17] $\alpha = 5$ and $\beta = .7$ were found to work well. We note that in our testing with jittered data we found little difference in performance, and in general used (35).

With $h_0(x)$ in hand, we replace $\delta_0(x)$ with the approximation $\sum_{|k|\leq N} \hat{h}(\lambda_k)\tilde{\varphi}_{M,k}(x)$ in (33), where $\hat{h}(\lambda_k)$ are the Fourier coefficients of $h_0(x)$. This leads to the explicit expression for σ:

$$\sigma(k) = \frac{\hat{h}(\lambda_k)}{\hat{r}(\lambda_k)}, \quad k = -N, \cdots, N.$$

In our experiments we used (35) which explicitly yields $\sigma(k) = \frac{2i\pi\lambda_k}{2N+1}$ for each $k = -N, \cdots, N$.

4.1.2. *Sparsity forward model for edge detection.* Now that we have an explicit epression for σ, we are able to derive a forward model for edge detection. Here we use the sparsity model developed in [17], which is based on the waveform matching idea developed in [34] for uniform Fourier data. This approach allows us to build a forward model for reconstructing $[f](x)$ using convex optimization on an ℓ_1-regularized cost function. From [17], for non-uniform Fourier data, the waveform kernel is given by

$$W^{\sigma}_{N(M)}(x) = \frac{1}{\gamma^{\sigma}_{N(M)}}\sum_{k=-M}^{M}\sigma_k\hat{r}(\lambda_k)\tilde{\varphi}_{N,k}(x) = \frac{1}{\gamma^{\sigma}_{N(M)}}\sum_{k=-M}^{M}\sum_{l=-N}^{N}\sigma_k\hat{r}(\lambda_k)b_{l,k}e^{i\pi lx},$$

where

$$\gamma^{\sigma}_{N(M)} = \sum_{k=-M}^{M}\sum_{l=-N}^{N} b_{l,k}\sigma_k\hat{r}(\lambda_k)$$

is the normalization constant. We then approximate

$$W^{\sigma}_{N(M)} * [f] \approx T^{\sigma}_{M}(f),$$

which is satisfied by holding

$$\frac{1}{\gamma^{\sigma}_{N(M)}} \sum_{k=-M}^{M} b_{l,k}\sigma_k \hat{r}(\lambda_k)\widehat{[f]}(l) \approx \sum_{k=-M}^{M} b_{l,k}\sigma_k \hat{f}(\lambda_k), \quad l = -N, \ldots, N.$$

Therefore the waveform kernel approach promotes the construction of the model

$$\frac{1}{\gamma^{\sigma}_{N(M)}}(\mathbf{B}(\sigma \cdot \hat{\mathbf{r}})) \cdot (\mathbf{F}[f]) \approx \mathbf{B}(\sigma \cdot \hat{\mathbf{f}})$$

where $\mathbf{B}$ is the matrix with entries $b_{l,k}$ given in (27), $\cdot$ denotes elementwise multiplication, and $\mathbf{F}$ is the Fourier transform. This model can be written simply as

$$\mathbf{A}[f] \approx \mathbf{y} \tag{36}$$

where

$$\mathbf{A} = \frac{1}{\gamma^{\sigma}_{N(M)}} \text{diag}\left[\mathbf{B}(\sigma \cdot \hat{\mathbf{r}})\right] \mathbf{F} \quad \text{and} \quad \mathbf{y} = \mathbf{B}(\sigma \cdot \hat{\mathbf{f}}).$$

We now have all of the ingredients needed to approximate $[f](x)$ on a set of grid points. The following equation combines the model in (36) with ℓ_1 regularization to construct an approximation to $[f](x)$ as the solution to the optimization problem (see Algorithm 4 in [17]) which is given by

$$\mathbf{g}^* = \arg\min_{\mathbf{g}} \left(||\mathbf{A}\mathbf{g} - \mathbf{y}||_2 + \mu ||\mathbf{g}||_1 \right). \tag{37}$$

The cost function is ℓ_1-regularized to promote the sparsity of the edge function vector $\mathbf{g}$, with $\mu > 0$ being the regularization parameter. Figure 5 demonstrates the use of (37) on $f_1(x)$ and $f_2(x)$.

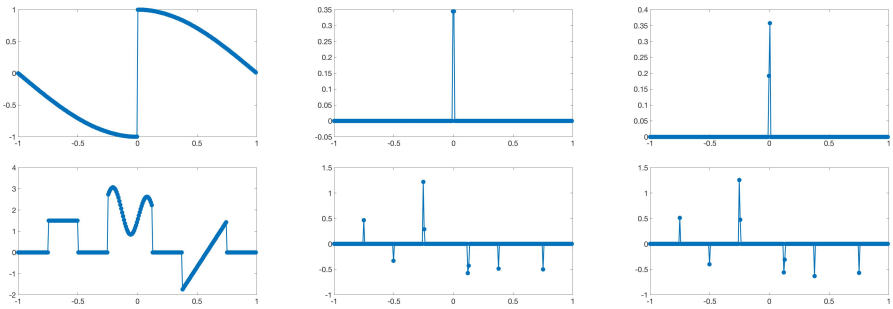

FIGURE 5. (Left) $f_1(x)$ and $f_2(x)$; (Middle) Jump function approximations using (37); (Right) Jump function approximations with additive noise starting from (3) with SNR = 20 dB. Here we use 257 reconstruction points, 257 jittered Fourier modes, and regularization parameter $\mu = 1$.

In two dimensions, the jump functions in the x and y directions may be approximated by the respective solutions to the optimization problems

$$\begin{aligned} \mathbf{g}^*_x &= \arg\min_{\mathbf{g}} ||\mathbf{A}\mathbf{g} - \mathbf{y}||_2 + \mu ||\mathbf{g}||_1 \\ \mathbf{g}^*_y &= \arg\min_{\mathbf{g}} ||\mathbf{g}\mathbf{A}^{\mathbf{T}} - \mathbf{y}||_2 + \mu ||\mathbf{g}||_1, \end{aligned} \tag{38}$$

where $\sigma_x(\mathbf{k}) = \frac{2i\pi\lambda_{k_1}}{2M+1}$ and $\sigma_y(\mathbf{k}) = \frac{2i\pi\lambda_{k_2}}{2M+1}$, $k_1, k_2 = -M, \cdots, M$, are used to create the corresponding $\mathbf{A}$ and $\mathbf{y}$ for each direction. Figure 6 shows the result for approximating $[f_3](x, y)$.

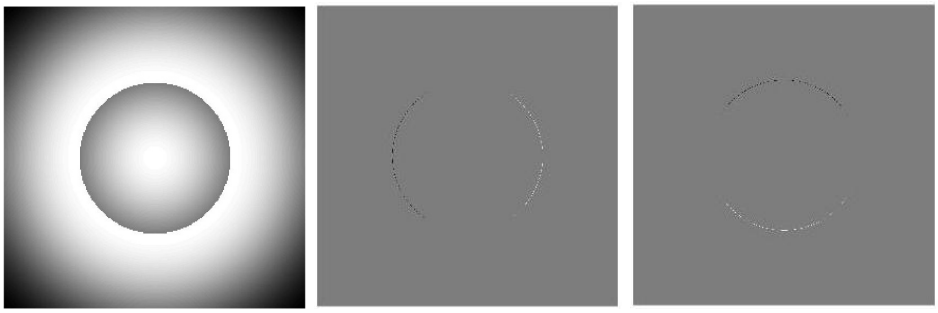

FIGURE 6. Edge detection (center and right) in the x and y directions using (38) for $f_3(x, y)$ (left). Here we use 257×257 jittered Fourier modes, 257×257 reconstruction points, and regularization parameter $\mu = 1$.

Of course, edge detection from non-uniform Fourier data is itself an important tool in identifying physical structures in images or signals. Thus a potential byproduct of our new reconstruction algorithm is that it produces an accurate *edge map* that is computed en route to reconstruction. Moreover, it can also act as a cross-validation for the reconstruction.

4.2. **Edge-adaptive ℓ_2 regularization image reconstruction algorithm.** Once the edge map is formed via (37), (38), or with another method specific to the user's application, the edge-adaptive ℓ_2 regularization reconstruction method begins with the creation of a mask, or weighting matrix. The edge detection process described above is critical in determining the weighting mask for the regularization term, and is even more important when the given data are noisy. The edge mask will enable us to adapt the regularization parameter to weight relatively lightly on fidelity in smooth regions and heavily on fidelity in edge regions where we have more confidence that there are non-zero values in the gradient domain. The mask creation process is described below, with the first step detailed in Algorithm 3 for the one-dimensional case and Algorithm 4 for the two-dimensional case.

For the purposes of mask creation, we need only a binary edge map indicating whether or not an edge has been detected at a given grid point. In one dimension, this is easily generated as

$$\mathbf{y}_j = \begin{cases} 1 & \text{if } |\mathbf{g}_j^*| > \tau \\ 0 & \text{otherwise,} \end{cases} \qquad j = -J, \ldots, J. \tag{39}$$

Here $\mathbf{g}^*$ is determined from (37) and τ is a user-defined threshold which is explained below. In two dimensions, we generate the binary edge maps as

$$\mathbf{x}_{i,j} = \begin{cases} 1 & \text{if } |(\mathbf{g}_x^*)_{i,j}| > \tau \\ 0 & \text{otherwise,} \end{cases} \qquad i, j = -J, \ldots, J.$$

and

$$\mathbf{y}_{i,j} = \begin{cases} 1 & \text{if } |(\mathbf{g}_y^*)_{i,j}| > \tau \\ 0 & \text{otherwise,} \end{cases} \qquad i, j = -J, \ldots, J.$$

where $\mathbf{g}_x^*$ and $\mathbf{g}_y^*$ are determined from (38).

Because the mth order PA transform will be used in the regularization term for the reconstruction procedure, the corresponding regularization mask must include a stencil surrounding each nonzero value. This is because the PA transform forms an oscillatory response in the $m+1$ point stencil surrounding the nonzero value. For example, when using L^3 in (11), there is an oscillatory response in 4 points surrounding a nonzero value. It is possible to simply define the mask to include the $m+1$ points in either direction of the found edge. However, this would require some bookmarking. Hence instead we use a second process to recover the response of applying L_m to the binary edge map. The latter steps in Algorithms 3 and 4 demonstrate this process.

Algorithm 3 Mask creation in one dimension

1: Starting from Fourier data as in (3), reconstruct the jump function, $[f]$, using equation (37) as $\mathbf{g}^*$.
2: For each index $j = -J, \dots, J$ such that $|\mathbf{g}_j^*| > \tau$, set $\mathbf{y}_j = 1$. Else, $\mathbf{y}_j = 0$.
3: For each index $j = -J, \dots, J$ such that $|(L^m \mathbf{y})_j| > 0$, set $\mathbf{z}_j = 0$. Else, $\mathbf{z}_j = 1$. The mask is $M = \text{diag}(\mathbf{z})$.

Algorithm 4 Mask creation in two dimensions

1: Starting from Fourier data as in (4), reconstruct the jump functions in the x and y directions using equations (38) as $\mathbf{g}_x^*$ and $\mathbf{g}_y^*$.
2: For each index $i, j = -J, \dots, J$ such that $|(\mathbf{g}_x^*)_{i,j}| > \tau$, set $\mathbf{x}_{i,j} = 1$. Else, $\mathbf{x}_{i,j} = 0$.
3: For each index $i, j = -J, \dots, J$ such that $|(\mathbf{g}_y^*)_{i,j}| > \tau$, set $\mathbf{y}_{i,j} = 1$. Else, $\mathbf{y}_{i,j} = 0$.
4: For each index $i, j = -J, \dots, J$ such that $|(L^m \mathbf{x})_{i,j}| > 0$, set $M_{i,j}^x = 0$. Else, $M_{i,j}^y = 1$. The x direction mask is the matrix M^x.
5: For each index $i, j = -J, \dots, J$ such that $|(\mathbf{y}(L^m)^T)_{i,j}| > 0$, set $M_{i,j}^y = 0$. Else, $M_{i,j}^y = 1$. The y direction mask is the matrix M^y.

Remark 2. A thresholding parameter τ has been introduced in Algorithms 3 and 4. However, our experiments demonstrate that the method is robust for a wide range of τ. Indeed, for the example in Figure 5, (nearly) exact edge locations were recovered for $\tau \in [0, \min [f](x))$. This robustness is due to the fact that most values resulting from (37) and (38) are indeed zero as these equations encourage sparsity. In this way, τ simply propagates the edge detection result to the mask. Although τ is function dependent, as a heuristic for choosing τ, we propose that it should be inversely proportional to the number of grid points (e.g. $\frac{1}{2J+1}$), as with better resolution we should be able to detect smaller magnitude jumps and include them in the mask. We can further specify τ in terms of PA order m, as was discussed in Theorem 3.3 in [2]. However, when noise is present, such precise thresholding based on grid resolution and PA order is not as effective, as ultimately the threshold will depend on the signal to noise ratio. In our experiments we typicaly chose τ as proportional to $\frac{1}{2J+1}$.

To be clear, Algorithms 3 and 4 create binary masks that dictate suspected nonzero locations in the gradient domain of the signal or image to be reconstructed. At this stage, the mask has been formed. The next step is to use the mask to target regularization only in smooth regions, recalling that earlier we defined smooth regions to be where the mask is nonzero. This simply involves weighting the regularization term with the mask. Put another way, our method uses regularization on targeted areas that we actually expect to be zero in the gradient domain. Therefore, with a properly chosen mask, it is appropriate to regularize using the ℓ_2 norm since we no longer require a sparsity-encouraging norm. Using ℓ_2 makes the algorithm much more cost efficient. In edge regions where the mask is zero, our method relies solely on the fidelity term. Reconstruction of both the smooth and edge regions are performed by minimizing a single cost function.

As a further justification for using ℓ_2, observe that minimizing $||ML^m\mathbf{g}||_2^2$ in is akin to setting up the usual sparsity constraint, which requires $L^m g$ to have only a few non-zero values, or more precisely, values above a chosen threshold. However, by employing the mask, we expect *no* non-zero values, making the ℓ_2 norm appropriate. The result is Algorithm 5 for one-dimensional signals and Algorithm 6 details the algorithm for two-dimensional functions and images.

Algorithm 5 Edge-adaptive image reconstruction in one dimension

1: Construct the mask, M, using Algorithm 3.
2: The edge-adaptive ℓ_2 regularization image reconstruction is the solution to the optimization problem,

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2^2 + \lambda ||ML^m\mathbf{g}||_2^2 \right), \tag{40}$$

where $\lambda > 0$ is the regularization parameter.

We note that (40) has a closed form solution

$$\mathbf{f}^* = \left(\mathcal{F}^T\mathcal{F} + \lambda (L^m)^T M L^m\right)^{-1} \mathcal{F}^T\hat{\mathbf{f}}. \tag{41}$$

This closed form may be valuable in some contexts. As the size of the problem increases, however, the inversion in (41) becomes more computationally expensive, so in Section 5 we take advantage of the conjugate gradient descent method [22].

Algorithm 6 Edge-adaptive image reconstruction in two dimensions

1: Construct the masks, M^x and M^y, using Algorithm 4.
2: The edge-adaptive ℓ_2 regularization image reconstruction is the solution to the optimization problem,

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2^2 + \lambda \left(\sum_{i=-J}^{J} \sum_{j=-J}^{J} M^x_{i,j} (L^m\mathbf{g})^2_{i,j} + \sum_{i=-J}^{J} \sum_{j=-J}^{J} M^y_{i,j} (\mathbf{g}(L^m)^T)^2_{i,j} \right) \right), \tag{42}$$

where $\lambda > 0$ is the regularization parameter.

Observe that unlike the iterative weights described in Section 3, the weighting masks generated by Algorithms 3 and 4 are binary. As our numerical results will

show, accuracy is improved since in general we do not falsely identify smooth variation or noise as jumps like the IR methods described in Section 3. Our numerical results will also show improved resolution around jumps, better noise reduction in smooth areas (see Figure 17), and better efficiency including a justification of using the ℓ_2 norm.

5. **Numerical results.** In the numerical experiments that follow we compare the edge-adaptive ℓ_2 regularization image reconstruction given by Algorithms 5 and 6 to the iteratively reweighted method of Algorithm 1 and 2. We use the Split Bregman method, [21, 37] to implement the minimization step in Algorithms 1 and 2. We follow the recommendation in [7] in choosing the parameter ϵ to be slightly smaller than the expected nonzero magnitudes of $L^m\mathbf{f}$, since this will provide the necessary stability to correct for inaccurate coefficient estimates while still improving upon the unweighted TV algorithm. We note that in [9] it is shown that updating ϵ in each iteration yields superior results for the problem of sparse signal recovery. However, since this is not applicable for functions with more variation, we did not consider this adaptive approach. Algorithms 5 and 6, which only require ℓ_2-regularized minimization, are performed using conjugate gradient descent, [22]. In what follows we look at examples in both one and two dimensions and the results using these algorithms with different regularization parameters. We also vary the PA order m, add noise to the initial data, and limit the amount of initial data. We also compare with an ℓ_1 regularized version of Algorithm 5. In all cases we compare the accuracy and efficiency of each algorithm. Finally, we demonstrate the success of our new algorithm on synthetic aperture radar (SAR) data, [12].

One-dimensional test case. Figure 7 compares the results of Algorithm 1 and Algorithm 5 for $f_1(x)$ in (20), where the acquired data are 257 noise-free jittered Fourier samples given by (1). We computed the relative error,

$$RE = ||\mathbf{f}^* - \mathbf{f}||_2/||\mathbf{f}||_2, \tag{43}$$

for each algorithm, resulting in $RE = .0446$ using Algorithm 1 and $RE = .0155$ using Algorithm 5. In addition to improving the overall accuracy, it is evident that due to the precise jump identification yielded using (39), there is improved resolution and reduced error in the neighborhood of the jump.

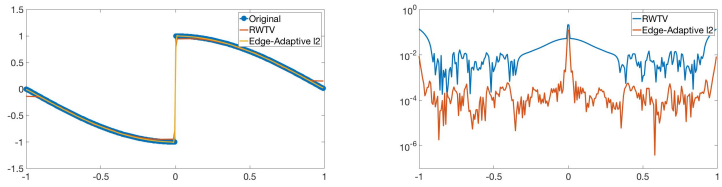

FIGURE 7. (Left) Comparison of Algorithms 1 and 5 on $f_1(x)$ using PA order $m = 1$ given 257 jittered Fourier samples reconstructed on 257 grid points; (Right) corresponding pointwise errors. For parameters, we use $\rho = 1$, $\mu = 1$, $\lambda = 1$, $\epsilon = 1.9$, $\ell_{max} = 25$, and threshold $\tau = 1/257$.

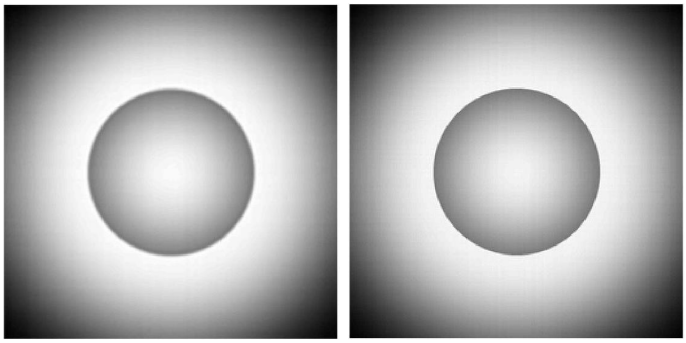

FIGURE 8. Comparison of Algorithms 2 (left) and 6 (right) for reconstructing $f_3(x, y)$ from 257×257 noise-free Fourier modes on 257×257 grid points. For parameters, we use PA order $m = 2$ due to the piecewise quadratic nature of the function, $\rho = .01$, $\epsilon = .9$, and $\ell_{max} = 5$, $\mu = .1$, $\tau = .025$, and $\lambda = 1$.

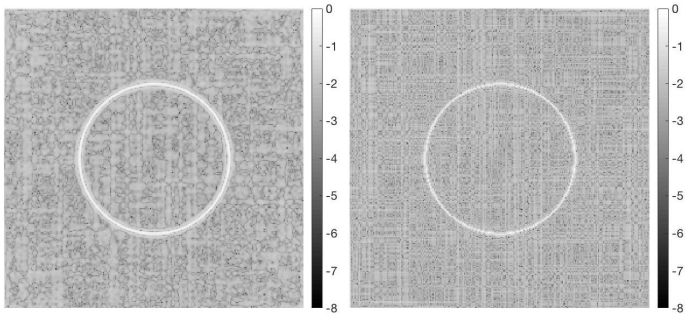

FIGURE 9. Comparison of errors using Algorithms 2 (left) and 6 (right) for reconstructing $f_3(x, y)$, using the same parameters as Figure 8.

Two-dimensional test case. Similar results are obtained in two dimensions, as is confirmed in Figures 8 and 9, which compare the results using Algorithms 2 and 6 for $f_3(x, y)$ given in (22). The data acquired are 257^2 noise-free jittered Fourier samples given by (4). It is evident that Algorithm 6 yields both better overall accuracy in terms of relative error, $RE = .0414$ for Algorithm 6 versus $RE = .0616$ for Algorithm 2, as well as improved resolution near the edges of the image, exhibited by the smaller white ring in its error plot.

Robustness of regularization parameter. Choosing the regularization parameter for the minimization step of optimization-based reconstruction methods, e.g. as in (10), is typically difficult and problem-dependent, [28], yet crucial to the success of the algorithm. Using the edge-adaptive ℓ_2 method, we observe a robustness with respect to the choice of this parameter. This is shown in Figure 10 (right), which displays the pointwise error plots comparing Algorithm 1 (RWTV) and Algorithm 5 for reconstructing $f_1(x)$ in (20) for various values of λ. Observe that our edge-adaptive ℓ_2 method outperforms the RWTV reconstruction for a wide range of λ.

Such robustness is critical since in many applications reliable ground truth information is not available. In terms of relative error, Algorithm 1 yielded $RE = .0446$, while even in the worst case, $\lambda = 100$, Algorithm 5 produced $RE = .0478$. Moreover, it is evident that for all choices of λ, there is improved resolution using our algorithm in neighborhoods of the jump. These results are particularly impressive when compared with the robustness of Algorithm 1 with respect to the regularization parameter ρ as seen in Figure 10 (left). There we see that the accuracy varies strongly with ρ, in particular around the jump. Regardless of the choice of λ, Algorithm 5 outperformed Algorithm 1, especially near the jump.

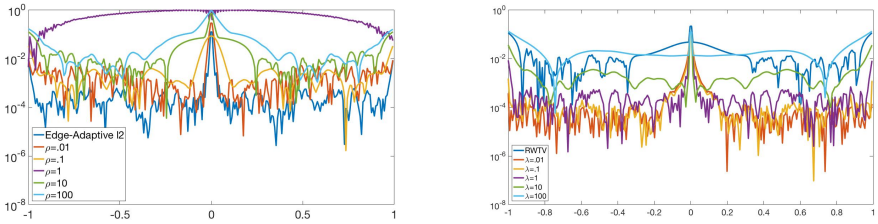

FIGURE 10. Pointwise error plot comparisons between Algorithm 1 and Algorithm 5 for $f_1(x)$ given 257 jittered Fourier samples reconstructed on 257 grid points. (Left) Parameters $m = 1$, $\ell_{max} = 25$, $\epsilon = 1.9$, $\tau = 1/257$, $\lambda = 1$, and vary $\rho = .01, .1, 1, 10, 100$; (Right) Parameters $m = 1$, $\rho = 1$, $\ell_{max} = 25$, $\epsilon = 1.9$, $\mu = 1$, $\tau = 1/257$, and vary $\lambda = .01, .1, 1, 10, 100$.

Comparison of PA order. Recall that the polynomial annihilation (PA) method annihilates polynomials of degree $m - 1$ in smooth regions, with $m = 1$ being an ideal choice for piecewise constant functions and $m = 2$ for piecewise linear functions. When functions and images contain more variation, such as for $f_2(x)$ in (21), choosing $m > 1$ may yield better results. Figure 11 compares the results using $m = 1, 2, 3$ for reconstructing $f_2(x)$. In each case, overall improved accuracy is evident with smaller error around jumps when using Algorithm 5.

One-dimensional examples with noise. Noise in the data acquisition process of imaging systems has the potential to seriously degrade the quality of a reconstruction. Figure 12 compares each algorithm for both $f_1(x)$ and $f_2(x)$ when our data consists of 257 noisy jittered Fourier samples, (3). Here the noise is assumed to be zero-mean complex Gaussian. While the edge-adaptive ℓ_2 no longer provides significant improvement in the overall error, it is still evident that the functions are resolved better in the neighborhoods of the jumps.

Limited data (compressed sensing) example. To test our algorithm's performance when starting from limited data, we consider the Shepp-Logan phantom, [33], shown in Figure 13. In this experiment, we randomly select just part of the initial Fourier data to use. Starting from 257×257 jittered Fourier modes as in (4), Figures 14, 15 and 16 respectively show the results using roughly a fourth, half, and three fourths of these modes which are chosen uniformly at random. It is evident that the edge-adaptive algorithm is particularly effective when the edges are close

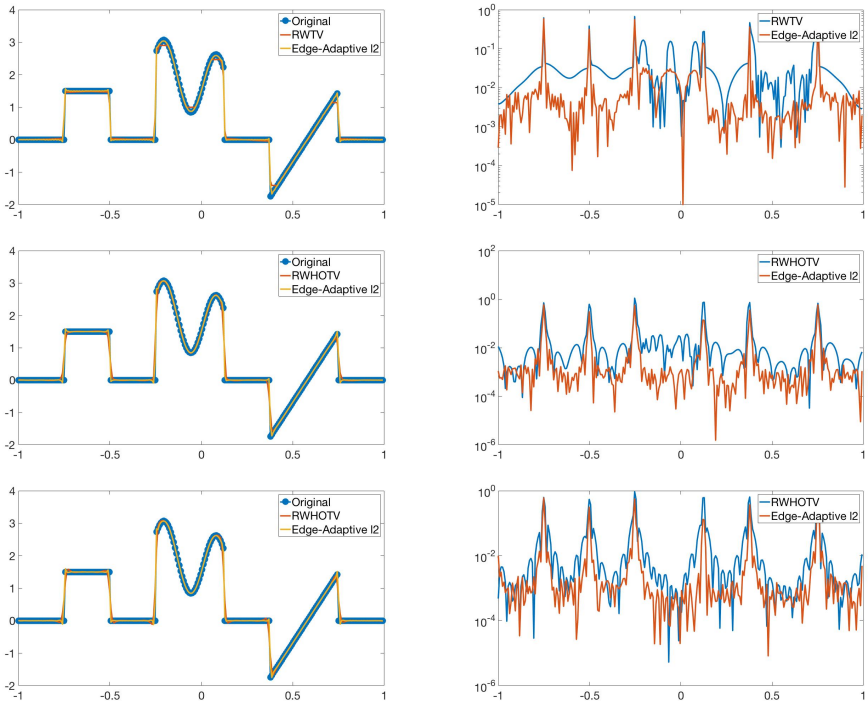

FIGURE 11. Comparison of Algorithms 1 and 5 on $f_2(x)$ using (top) $m = 1$ (middle) $m = 2$ and (bottom) $m = 3$ given 257 jittered Fourier samples reconstructed on 257 grid points. Left are the reconstructions, right are the corresponding pointwise errors. For parameters, we use $\rho = 1$, $\ell_{max} = 25$, $\epsilon = 2.9$, $\mu = 1$, $\tau = 1/257$, and $\lambda = 1$.

together, that is, it appears in general to have better resolution properties. Further theoretical and numerical study is needed to determine precisely the maximum compression ratio achievable by the edge adaptive ℓ_2 method.[6]

Synthetic aperture radar (SAR) example. As a final example, we consider the synthetic aperture radar (SAR) phase history data of a vehicle given in [12]. SAR is an all weather, night or day imaging modality whereby an image is reconstructed from electromagnetic scattering data. In SAR we assume only a sparse number of isotropic point scatterers, so the standard ℓ_1 regularized reconstruction solves

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2^2 + \lambda ||\Theta^* \mathbf{g}||_1 \right) \tag{44}$$

where Θ^* is a diagonal phase extraction matrix yielding $\Theta^* \mathbf{f} \approx |\mathbf{f}|$. This is needed because the phase of $\mathbf{f}$ is not sparse. (See e.g. [31] for details on the construction of

[6]We note that typically Shepp Logan phantom reconstructions using compressive sensing algorithms come from either uniform or radial discrete Fourier data, [4, 7], which we are not considering here.

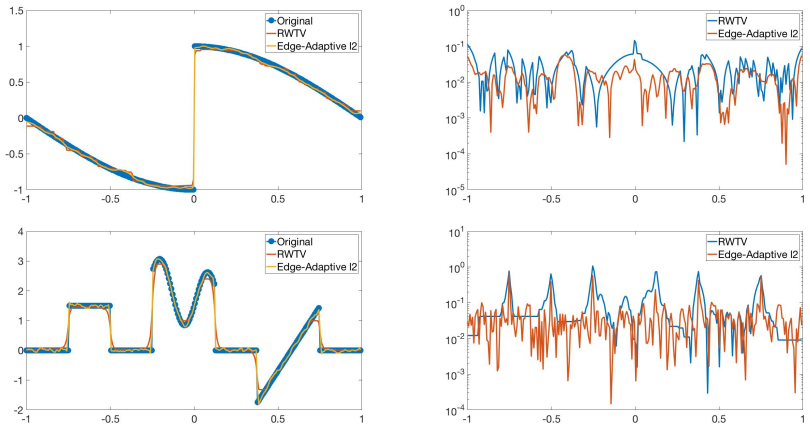

FIGURE 12. (Left) Comparisons of Algorithms 1 and 5 on $f_1(x)$ and $f_2(x)$ given 257 noisy jittered Fourier samples. $SNR = 15$ for $f_1(x)$ and SNR $= 20$ for $f_2(x)$. In both cases we use PA order $m = 1$ and reconstruct on 257 grid points; (Right) corresponding pointwise errors. For $f_1(x)$ we use parameters $\rho = 1$, $\epsilon = 1.9$, $\ell_{max} = 25$, $\mu = 1$, $\tau = 1/257$, and $\lambda = 1$. For f_2 we use parameters $\rho = 1$, $\epsilon = 2.9$, $\ell_{max} = 25$, $\mu = 1$, $\tau = 1/257$, and $\lambda = 1$.

FIGURE 13. 257×257 pixel Shepp-Logan phantom.

Θ.) The edge-adaptive ℓ_2 method for this application is then

$$\mathbf{f}^* = \arg\min_{\mathbf{g}} \left(||\mathcal{F}\mathbf{g} - \hat{\mathbf{f}}||_2^2 + \lambda ||M\mathbf{g}||_2^2 \right) \tag{45}$$

where $M = M^x + M^y$ is the mask found through Algorithm 4. Since we are now using ℓ_2 regularization, the phase extraction matrix Θ^* is unnecessary. SAR data have a significant amount of noise, [14]. Nevertheless we are able to locate the edges with relatively high confidence. Algorithm 6 is particularly effective in this case because we can heavily penalize the regularization term, and for our experiments we chose $\lambda = 100$. Figure 17 compares the results reconstructing via equation (44) and Algorithm 6 for the given SAR data set.

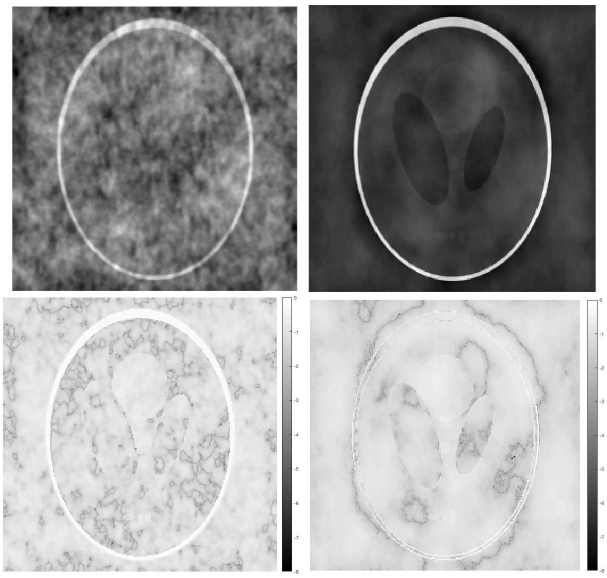

FIGURE 14. (Top) Reconstruction on 257×257 grid points of the Shepp-Logan phantom using Algorithms 2 (left) and 6 (right) with PA order $m = 1$ from 129^2 Fourier coefficients randomly chosen from a grid of 257×257. (Bottom) respective pointwise errors. For parameters, we use $\rho = .01$, $\epsilon = .9$, $\ell_{max} = 5$, $\mu = .01$, $\tau = 0.1$, $\lambda = .1$. The relative error using Algorithm 2 was $RE = .7160$, while Algorithm 6 yielded $RE = .4873$.

Efficiency of edge-adaptive ℓ_2 minimization. Our new edge-adaptive ℓ_2 method is more efficient than Algorithm 2 in all experiments. This is to be expected since in general ℓ_2-regularized problems are much easier to solve than ℓ_1-regularized minimizations. More specifically, the major cost for both Algorithms 2 and 6 is a conjugate gradient (CG) solve. Algorithm 6 requires only one CG solve, while Algorithm 2 uses the split Bregman method, [21], which requires multiple CG solve for each reconstruction iteration. Therefore if the convergence tolerance for the CG step is the same, Algorithm 6 will be more efficient than Algorithm 2 on the order of the number of CG per iteration times the the number of iterations. The number of CG solves per iteration can range from $\mathcal{O}(10)$ to $\mathcal{O}(1000)$, and the number of iterations can be as many as 10.

Data size	Algorithm 2	Algorithm 6
129×129	4 mins 10 secs	5.6 secs
257×257	13 mins 2 secs	22 secs
513×513	49 mins 26 secs	1 min 33 secs
1025×1025	3 hours 16 mins	6 mins 28 secs

TABLE 1. Run time comparison between Algorithms 2 and 6 for reconstructing $f_3(x, y)$. We used $\ell_{max} = 5$. The run time includes the time to perform Algorithm 4.

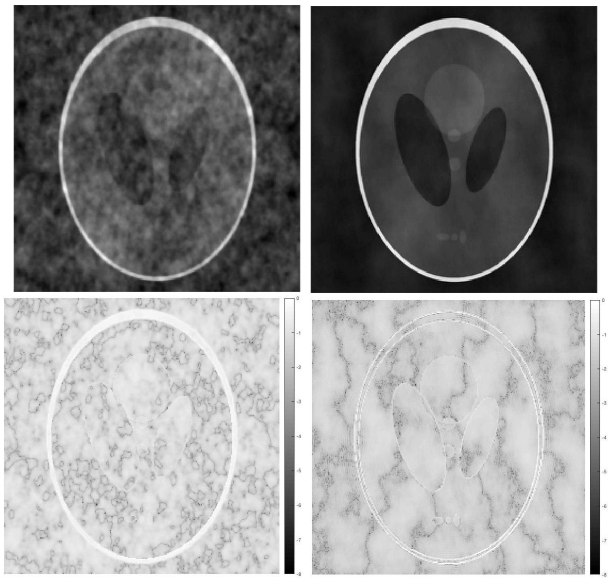

FIGURE 15. (Top) Reconstruction on 257×257 grid points of the Shepp-Logan phantom using Algorithms 2 (left) and 6 (right) with PA order $m = 1$ from 181^2 Fourier coefficients randomly chosen from a grid of 257×257. (Bottom) respective pointwise errors. For parameters, we use $\rho = .01$, $\epsilon = .9$, $\ell_{max} = 5$, $\mu = .01$, $\tau = 0.1$, $\lambda = .1$. The relative error using Algorithm 2 was $RE = .5180$, while Algorithm 6 yielded $RE = 0.3259$.

Table 1 compares runtimes[7] for Algorithm 2 using $\ell_{max} = 5$ and Algorithm 6 including the mask generation in Algorithm 4 for $f_3(x, y)$. For smaller images, e.g. those reconstructed on a 129×129 pixel grid given 129×129 Fourier samples, the runtime for Algorithm 6 is in seconds, compared to minutes for Algorithm 2. Note that this means that Algorithm 6, including edge detection, is faster than even a single iteration of Algorithm 2. These gains are even more significant as the images increase in size. For example, given 1025×1025 Fourier samples reconstructed on 1025×1025 grid points, our new algorithm computes the results in about 6 and a half minutes, while Algorithm 2 took over 3 hours, which is over 30 minutes per iteration. We note that we did not implement accelerated homotopy-based algorithms for reweighted ℓ_1 methods as in [3], which may increase computational speed. In addition, the iteratively reweighted least squares method developed in [9] would also run more efficiently, since it also uses an ℓ_2 norm in the regularization. However, we would expect this method to also suffer from the same inaccuracies that arise from iteratively finding edges.

Edge-adaptive ℓ_1 minimization. Finally, we consider Algorithms 5 and 6 but instead with ℓ_1 regularization rather than ℓ_2. While efficiency is obviously compromised, using ℓ_1 may sometimes be appropriate, especially when only a partial edge map is detected. In this case the rest of the supposed smooth region of the

[7] All computations were performed on a MacBook Air with a 1.7 GHz Intel Core i5 processor and 4 GB of memory.

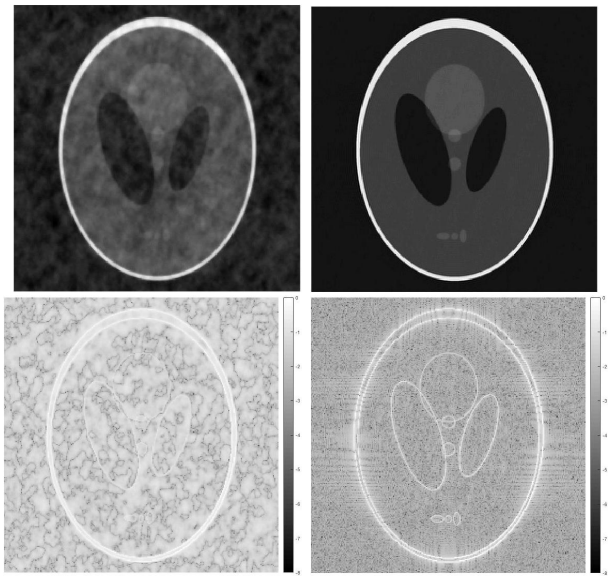

FIGURE 16. (Top) Reconstruction on 257×257 grid points of the Shepp-Logan phantom using Algorithms 2 (left) and 6 (right) with PA order $m = 1$ from 225^2 Fourier coefficients randomly chosen from a grid of 257×257. (Bottom) respective pointwise errors. For parameters, we use $\rho = .01$, $\epsilon = .9$, $\ell_{max} = 5$, $\mu = .01$, $\tau = 0.1$, $\lambda = .1$. The relative error using Algorithm 2 was $RE = .3458$, while Algorithm 6 yielded $RE = .2930$.

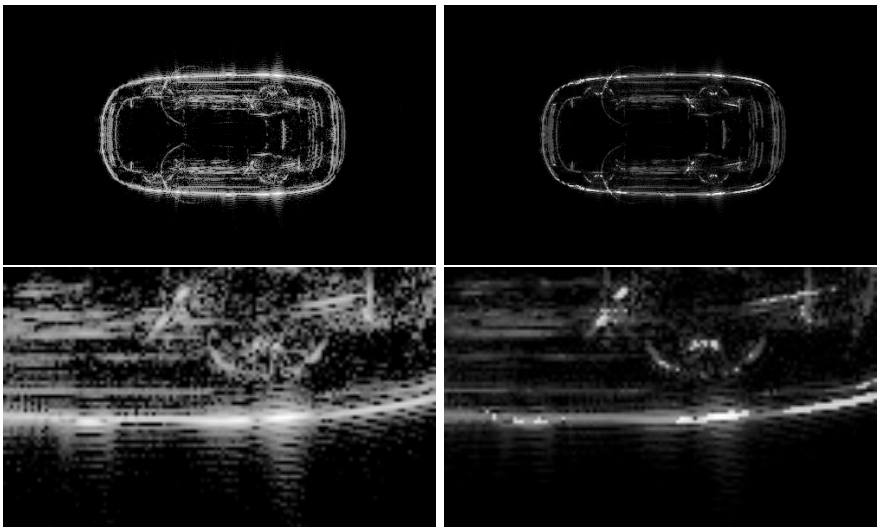

FIGURE 17. (Top) Reconstruction comparison of SAR vehicle data using (left) equation (44) and (right) Algorithm 6. (Bottom) A close up of the lower right tire.

image may also have a sparse gradient, and so a sparsity-encouraging ℓ_1 norm would be better suited. A comparison is shown in Figure 18 showing the error plots for reconstructions of $f_2(x)$ from jittered Fourier data with and without added noise. The relative errors for the noise-free reconstruction error plots are nearly equal with the ℓ_1 method achieving $RE = .0641$ and the ℓ_2 having $RE = .0640$. When noise is added, the ℓ_2 method slightly outperforms with $RE = .0692$ compared with $RE = .0706$ for ℓ_1.

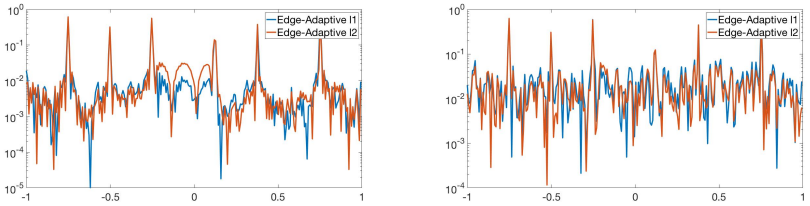

FIGURE 18. Comparison of edge-adaptive ℓ_1 and edge-adaptive ℓ_2 methods on $f_2(x)$ using 257 Fourier coefficients. (left) error plot from reconstructions with noise-free data. (right) error plot from reconstruction with 20 dB zero-mean Gaussian noise added to the data.

6. **Conclusion.** The edge-adaptive ℓ_2 regularization image reconstruction method introduced in this paper compares favorably in terms of image quality, sharpness around jumps, and noise reduction to ℓ_1 based iteratively reweighted (HO)TV regularization reconstructions. It is also more efficient, requiring just a single ℓ_1 minimization solution for edge detection that only needs to be performed once for an image. The method is flexible in that edges can be detected using any method, and if the edges are already known from some other experiment they can directly be used in our algorithm. After the edge detection, we can rely on faster conjugate gradient descent methods to solve the easier ℓ_2 minimization problem. For some applications it may be useful to use the edge map produced in Algorithm 4 as a cross-validation of the image. The results for compressed imaging are promising, although more work is needed to determine how much compression is possible. Future investigations will also include a variable (rather than binary) map, which may be important when the intensity of the images vary widely in scale. Moreover, this will allow us to generalize our technique to any problems for which separation of scales may be advantageous, that is, not just for identifying edges but perhaps also textures and other singularities. To this end, recent work in [32] on weighted ℓ_p regularization methods might be useful. We also will extend our new algorithm to multi-measurement vector (MMV) applications, as the efficiency gained by our method would be even more significant in this case. Finally, we will test our method on other types of acquired data as well as other sparsifying transform operators, such as wavelets, which may be advantageous in some applications.

REFERENCES

[1] R. Archibald, A. Gelb and R. B. Platte, Image reconstruction from undersampled Fourier data using the polynomial annihilation transform, *Journal of Scientific Computing*, **67** (2016), 432–452.

[2] R. Archibald, A. Gelb and J. Yoon, Polynomial fitting for edge detection in irregularly sampled signals and images, *SIAM Journal on Numerical Analysis*, **43** (2005), 259–279.
[3] M. S. Asif and J. Romberg, Fast and accurate algorithms for re-weighted ℓ_1-norm minimization, *IEEE Transactions on Signal Processing*, **61** (2013), 5905–5916.
[4] E. J. Candès, J. Romberg and T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, *IEEE Transactions on Information Theory*, **52** (2006), 489–509.
[5] E. J. Candes, J. K. Romberg and T. Tao, Stable signal recovery from incomplete and inaccurate measurements, *Communications on Pure and Applied Mathematics*, **59** (2006), 1207–1223.
[6] E. J. Candes and T. Tao, Near-optimal signal recovery from random projections: Universal encoding strategies?, *IEEE Transactions on Information Theory*, **52** (2006), 5406–5425.
[7] E. J. Candes, M. B. Wakin and S. P. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization, *Journal of Fourier Analysis and Applications*, **14** (2008), 877–905.
[8] T. Chan, A. Marquina and P. Mulet, High-order total variation-based image restoration, *SIAM Journal on Scientific Computing*, **22** (2000), 503–516.
[9] R. Chartrand and W. Yin, Iteratively reweighted algorithms for compressive sensing, in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, IEEE, 2008, 3869–3872.
[10] I. Daubechies, R. DeVore, M. Fornasier and C. S. Güntürk, Iteratively reweighted least squares minimization for sparse recovery, *Communications on Pure and Applied Mathematics*, **63** (2010), 1–38.
[11] D. L. Donoho, Compressed sensing, *IEEE Transactions on Information Theory*, **52** (2006), 1289–1306.
[12] K. E. Dungan, C. Austin, J. Nehrbass and L. C. Potter, Civilian vehicle radar data domes, in *Algorithms for Synthetic Aperture Radar Imagery XVII*, International Society for Optics and Photonics, **7699** (2010), 76990P.
[13] A. Dutt and V. Rokhlin, Fast Fourier transforms for nonequispaced data, *SIAM Journal on Scientific Computing*, **14** (1993), 1368–1393.
[14] G. Franceschetti and R. Lanari, *Synthetic Aperture Radar Processing*, CRC press, 2018.
[15] A. Gelb and D. Cates, Detection of edges in spectral data iii–refinement of the concentration method, *Journal of Scientific Computing*, **36** (2008), 1–43.
[16] A. Gelb and G. Song, A frame theoretic approach to the nonuniform fast Fourier transform, *SIAM Journal on Numerical Analysis*, **52** (2014), 1222–1242.
[17] A. Gelb and G. Song, Detecting edges from non-uniform Fourier data using Fourier frames, *Journal of Scientific Computing*, **71** (2017), 737–758.
[18] A. Gelb and E. Tadmor, Detection of edges in spectral data, *Applied and Computational Harmonic Analysis*, **7** (1999), 101–135.
[19] A. Gelb and E. Tadmor, Detection of edges in spectral data ii. nonlinear enhancement, *SIAM Journal on Numerical Analysis*, **38** (2000), 1389–1408.
[20] A. Gelb and E. Tadmor, Adaptive edge detectors for piecewise smooth data based on the minmod limiter, *Journal of Scientific Computing*, **28** (2006), 279–306.
[21] T. Goldstein and S. Osher, The split bregman method for l1-regularized problems, *SIAM Journal on Imaging Sciences*, **2** (2009), 323–343.
[22] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Fourth edition. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, 2013.
[23] I. F. Gorodnitsky and B. D. Rao, Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm, *IEEE Transactions on Signal Processing*, **45** (1997), 600–616.
[24] L. Greengard and J.-Y. Lee, Accelerating the nonuniform fast Fourier transform, *SIAM Review*, **46** (2004), 443–454.
[25] W. Guo and W. Yin, Edge guided reconstruction for compressive imaging, *SIAM Journal on Imaging Sciences*, **5** (2012), 809–834.
[26] J.-Y. Lee and L. Greengard, The type 3 nonuniform fft and its applications, *Journal of Computational Physics*, **206** (2005), 1–5.
[27] H. Mansour and Ö. Yilmaz, Support driven reweighted? 1 minimization, in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, IEEE, 2012, 3309–3312.

[28] S. Osher, M. Burger, D. Goldfarb, J. Xu and W. Yin, An iterative regularization method for total variation-based image restoration, *Multiscale Modeling & Simulation*, **4** (2005), 460–489.
[29] D. Rosenfeld, New approach to gridding using regularization and estimation theory, *Magnetic Resonance in Medicine*, **48** (2002), 193–202.
[30] L. I. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D: Nonlinear Phenomena*, **60** (1992), 259–268.
[31] T. Sanders, A. Gelb and R. B. Platte, Composite sar imaging using sequential joint sparsity, *Journal of Computational Physics*, **338** (2017), 357–370.
[32] T. Scarnati, *Recent Techniques for Regularization in Partial Differential Equations and Imaging*, PhD thesis, Arizona State University, 2018.
[33] L. A. Shepp and B. F. Logan, The Fourier reconstruction of a head section, *IEEE Transactions on Nuclear Science*, **21** (1974), 21–43.
[34] W. Stefan, A. Viswanathan, A. Gelb and R. Renaut, Sparsity enforcing edge detection method for blurred and noisy Fourier data, *Journal of Scientific Computing*, **50** (2012), 536–556.
[35] Y. Wang and W. Yin, Sparse signal reconstruction via iterative support detection, *SIAM Journal on Imaging Sciences*, **3** (2010), 462–491.
[36] D. Wipf and S. Nagarajan, Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions, *IEEE Journal of Selected Topics in Signal Processing*, **4** (2010), 317–329.
[37] W. Yin, S. Osher, D. Goldfarb and J. Darbon, Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing, *SIAM Journal on Imaging Sciences*, **1** (2008), 143–168.

Received May 2018; revised March 2019.

E-mail address: victor.a.churchill.gr@dartmouth.edu
E-mail address: archibaldrk@ornl.gov
E-mail address: annegelb@math.dartmouth.edu