

## Edge Guided Reconstruction for Compressive Imaging\*

Weihong Guo<sup>†</sup> and Wotao Yin<sup>‡</sup>

**Abstract.** We propose EdgeCS—an edge guided compressive sensing reconstruction approach—to recover images of higher quality from fewer measurements than the current methods. Edges are important image features that are used in various ways in image recovery, analysis, and understanding. In compressive sensing, the sparsity of image edges has been successfully utilized to recover images. However, edge detectors have not been used on compressive sensing measurements to improve the edge recovery and subsequently the image recovery. This motivates us to propose EdgeCS, which alternatively performs edge detection and image reconstruction in a mutually beneficial way. The edge detector of EdgeCS is designed to faithfully return partial edges from intermediate image reconstructions even though these reconstructions may still have noise and artifacts. For complex-valued images, it incorporates joint sparsity between the real and imaginary components. EdgeCS has been implemented with both isotropic and anisotropic discretizations of total variation and tested on incomplete  $k$ -space (spectral Fourier) samples. It applies to other types of measurements as well. Experimental results on large-scale real/complex-valued phantom and magnetic resonance (MR) images show that EdgeCS is fast and returns high-quality images. For example, it exactly recovers the  $256 \times 256$  Shepp–Logan phantom from merely 7 radial lines (3.03%  $k$ -space), which is impossible for most existing algorithms. It is able to accurately reconstruct a  $512 \times 512$  MR image with 0.05 white noise from 20.87% radial samples. On complex-valued MR images, it obtains recoveries with faithful phases, which are important in many medical applications. Each of these tests took around 30 seconds on a standard PC. Finally, the algorithm is GPU friendly.

**Key words.** compressive sensing, edge detection, total variation, discrete Fourier transform, magnetic resonance imaging

**AMS subject classifications.** 65K10, 65F22, 65T50, 68U10, 90C25

**DOI.** 10.1137/110837309

### 1. Introduction.

**1.1. Background.** Compressive sensing (CS) (cf. the pioneering works [5, 10]) acquires a signal of interest, not by taking many uniform samples, but rather by measuring a few incoherent linear projections followed by an optimization-based reconstruction that exploits the sparsity or compressibility of the signal. For sparse and compressible signals, it requires far fewer linear measurements than dictated by the Nyquist–Shannon sampling theory [31]. This fact makes it very useful in reducing sensing cost in a variety of applications.

One of the early applications of CS and sparsity-based reconstruction is image reconstruction. For example, the Rudin–Osher–Fatemi (ROF) model [29] recovers clean images from

---

\*Received by the editors June 14, 2011; accepted for publication (in revised form) March 26, 2012; published electronically July 3, 2012.

<http://www.siam.org/journals/siims/5-3/83730.html>

<sup>†</sup>Department of Mathematics, Case Western Reserve University, Cleveland, OH 44106 ([weihong.guo@case.edu](mailto:weihong.guo@case.edu)).

<sup>‡</sup>Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005 ([wotao.yin@rice.edu](mailto:wotao.yin@rice.edu)). This author's work was supported in part by NSF CAREER award DMS-07-48839, ONR grant N00014-08-1-1101, and an Alfred P. Sloan Research Fellowship.

noisy ones by minimizing total variation, and the authors of [4] reconstruct a  $256 \times 256$  Shepp–Logan phantom from its incomplete Fourier measurements taken over 22 equally spaced radial lines. Sparsity in wavelet or tight frame representations is utilized to reconstruct natural images [10] from incomplete measurements. Sparse and redundant representations over a learned dictionary is deployed to remove white Gaussian noise [1]. To better sparsely represent signals, optimal sensing and sparsifying bases are learned simultaneously in [11]. Interested readers are referred to two recently published books [32, 12] for more details.

Edges are important features for object tracking and identification, image reconstruction, segmentation, and many other imaging tasks. Edges are usually sparser than images themselves, and the sparsity of edges has been widely exploited in image reconstruction. However, edges are not the only component in images. In the CS measurements, edge and nonedge components are linearly encoded altogether. Recovering just the edges is not simple. On the other hand, there exist very reliable edge detectors which effectively discriminate edges from other image components. However, these methods work on images instead of CS measurements. We believe that one does not need to first reconstruct an image from CS measurements before exploiting its edges. Instead, the two can be integrated. Therefore, we propose studying novel CS image reconstruction approaches that take advantages of both edge sparsity and edge detection. We hope the results of this paper persuade the reader that CS image reconstruction and edge detection benefit each other. Besides our earlier work [17], we have not seen other work on utilizing edge sparsity and edge detection in an alternative way. In the one-dimensional context, related existing work includes [39], which reconstructs one-dimensional signals through iterative support detection, and [37, 13], which assume in advance that partial support of the underlying signal is known.

This paper has the following two main contributions.

- EdgeCS, an edge guided CS image reconstruction scheme, is introduced. Given undersampled and/or noisy measurements, EdgeCS reconstructs images with fewer errors and artifacts than the state-of-the-art methods. Given measurements with a low sampling rate and high noise variance, a regular decoder usually fails to reconstruct meaningful images. Instead of discarding the failed reconstruction, partial edges are detected from it and then used to adjust sparsity constraints and iteratively improve results. The implementation is based on a fast alternating direction algorithm.
- To better recover complex-valued images, the joint sparsity between the real and imaginary components is utilized in the edge detector. Applied in magnetic resonance imaging (MRI), it is able to improve phase information restoration from incomplete noisy spectral measurements.

**1.2. Related work.** Compared to our earlier conference paper [17], improved partial edge detectors are developed, and the second contribution is novel.

The previous work [39] exploits binary reweighted  $\ell_1$  for recovering one-dimensional sparse signals. While it can be applied to recover the sparse wavelet coefficients of an image and hence the image itself, it can hardly take advantage of image edges to improve the recovery. On images, EdgeCS offers much better quality.

The rest of this paper is organized as follows. Section 2 presents the overall EdgeCS approaches for one-dimensional signals and two-dimensional real/complex-valued images, and

section 3 studies edge detection. Section 4 presents numerical implementation of EdgeCS and discusses parameter selections. Section 5 describes numerical simulations results. Finally, conclusions and discussions are given in section 6.

**2. EdgeCS for one-dimensional signals and two-dimensional images.** Our goal is to recover an image from very few measurements. When examining existing methods, we observed that although insufficient measurements lead to low-quality solutions, there exists useful information in these low-quality solutions. We tried to “recycle” such information and found that after the regularization term (e.g.,  $\ell_1$  or total variation (TV)) is properly updated (e.g., to weighted  $\ell_1$  or TV), the new solution can become significantly better. Motivated by these observations, this section studies how to effectively extract jump/edge information from low-quality solutions and use it to improve weighted TV-based reconstruction.

Starting from undersampled measurements with excessive noise, we alternate image reconstruction and edge detection to enhance CS reconstruction for one- and two-dimensional signals.<sup>1</sup> Section 2.1 gives a one-dimensional example, which connects the proposed work to a previous result. Section 2.2 focuses on two-dimensional images. Joint sparsity enhanced EdgeCS is explained in section 2.3.

**2.1. One-dimensional signals.** We begin our exposition with a simple demo of reconstructing a piecewise constant signal from its random measurements. The signal  $\bar{u}$  has  $n = 200$  entries and 25 randomly located jumps with standard Gaussian sizes. Let  $\Psi$  be a  $60 \times 200$  Gaussian random matrix and  $b := \Psi\bar{u}$ . Our goal is to recover  $\bar{u}$  from  $b$  and  $\Psi$ .

Define  $TV(u) = \sum_{i=1}^{n-1} |u_{i+1} - u_i|$ . We compare the solutions of TV minimization,

$$\min \{TV(u) : \Psi u = b\},$$

and weighted TV minimization,

$$(2.1) \quad \min \left\{ \sum_{i=1}^{n-1} g_i |u_{i+1} - u_i| : \Psi u = b \right\},$$

where weights  $g_i$  are iteratively learned through jump detection in the following algorithm.

**Algorithm: One-dimensional EdgeCS—Jump Guided TV Minimization.**

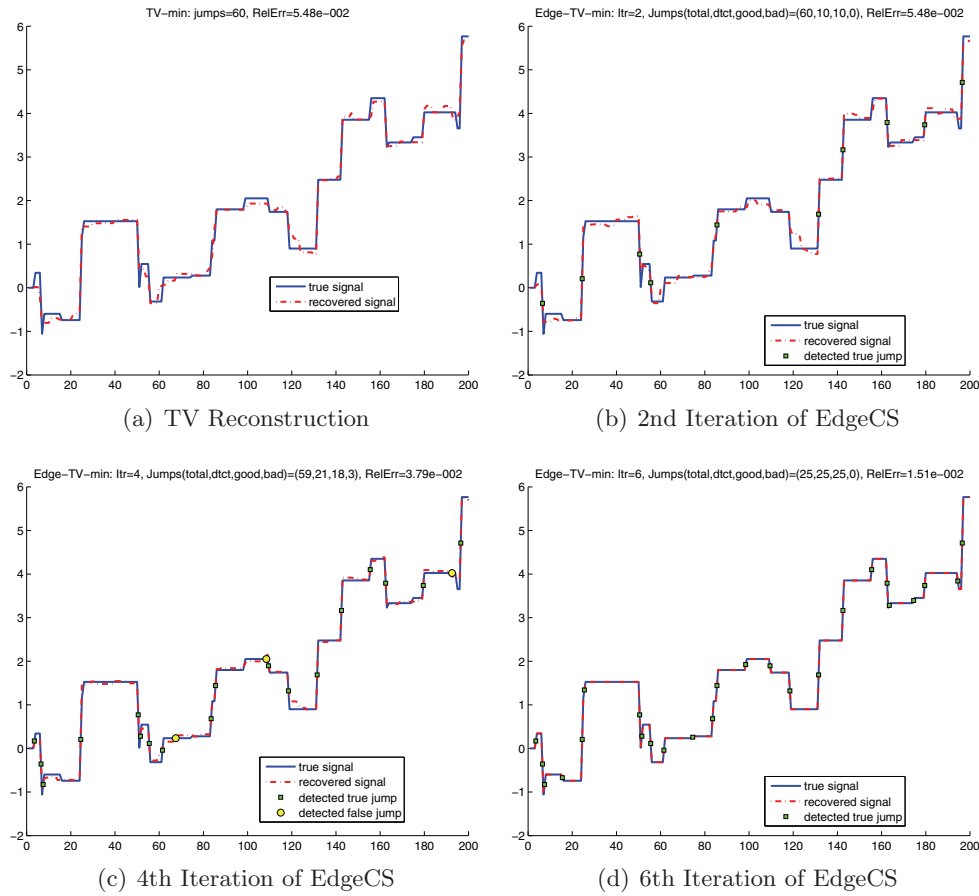
*Input:*  $\Psi, b, n$ .

1. *Iteration number*  $k \leftarrow 1$ ; *weights*  $g_i \leftarrow 1 \forall i$ ;
2. *While the stopping condition is not met, iterate*
  - (a) *Subproblem:*  $u^{(k)} \leftarrow \text{solve (2.1)}$ ;
  - (b) *Jump detection:*  $I^{(k)} \leftarrow \{i : |u_{i+1} - u_i| > 2^{-k} \max\{|u_{j+1} - u_j|, j = 1, \dots, n\}\}$ ;
  - (c) *Weight update in (2.1):*  $g_i \leftarrow 0 \forall i \in I^{(k)}$ ;  $g_j \leftarrow 1 \forall j \notin I^{(k)}$ ;
  - (d)  $k \leftarrow k + 1$ .

While direct TV minimization fails to recover  $\bar{u}$ , the algorithm One-dimensional EdgeCS recovered  $\bar{u}$  in merely six iterations. Since  $g_i \equiv 1$  initially, the first iteration coincided with

---

<sup>1</sup>Extensions to three- and higher dimensional signals are straightforward.



Itr $k$	Jumps				Relative Error $\frac{\ u^{(k)} - \bar{u}\ _2}{\ \bar{u}\ _2}$	
	Total	Detected	Good	Bad		
TV	1	60	6	6	0	5.48e-2
	2	60	10	10	0	4.71e-2
	3	59	14	14	0	3.79e-2
	4	59	21	18	3	2.86e-2
	5	59	23	21	2	1.51e-2
	6	25	25	25	0	1.82e-15

**Figure 1.** Comparison of TV reconstruction and EdgeCS (jump guided) reconstruction.

standard TV minimization. The solution of TV minimization and those at the end of the 2nd, 4th, and 6th iterations are depicted in Figure 1. Subfigures (b), (c), and (d) highlight the detected jumps including both the true and false ones. Relative errors are computed as  $\|u^{(k)} - \bar{u}\|_2 / \|\bar{u}\|_2$ . The quality of these jumps are given in quadruplets (total, dtct, good, bad), which are defined as follows:

- total: the total number of jumps in current  $u^{(k)}$ ;
- dtct: the number of detected jumps, equal to  $|I^{(k)}|$ ;

- good: the number of true jumps;
- bad: the number of false jumps.

The TV solution (1st iteration) roughly matches the true signal but misses many small jumps. It contains false jumps and artifacts that are relatively small in size, but most of its large jumps are preserved at their exact locations. Hence, thresholding at  $2^{-1} \max\{|u_{j+1} - u_j|\}$  identifies six of them (shown for  $k = 1$  in the table) with no false detections. By setting the corresponding  $g_i$  to 0 for the second iteration, a smaller reconstruction error is obtained. The improved solution of iteration 2 is then used for detecting more jumps. As EdgeCS iterates, the solutions improve and the thresholds reduce, so more jumps are detected. Note that at iteration 4, the detection includes three false jumps; however, since it also introduces four true jumps, iteration 5 yields a reduced reconstruction error. Generally, as long as false detections are relatively few, more detections lead to lower reconstruction errors. The detection at iteration 5, though including false jumps, has included enough true jumps to allow an exact reconstruction at iteration 6. The solution of iteration 6 exactly recovers all jumps and has a tiny error. All subproblems were solved with MATLAB linear programming solver “linprog” with the default parameters.

This demo shows that iterative jump detection can be very effective and indeed can help recover piecewise constant signals. This result can be partially explained by the analysis in [39], which uses, instead of the truncated TV here, truncated  $\ell_1$  minimization to recover one-dimensional sparse signals. Applying the analysis in [39] to iterative jump detection, we can conclude the following:

1. If the jumps are sparse enough, the recovery at the first iteration is exact; for fewer sparse jumps, more iterations and jump detections are needed.
2. For  $u^{(k)}$  to improve over the iterations, the jump detection at each iteration must discover new jumps, and they must include more correct ones than false ones. Detailed quantifications depend on the size and property of  $\Psi$ .
3. The total number of iterations is small (no more than nine empirically).

**2.2. Two-dimensional images.** Similar to reconstruction of one-dimensional signals, we use weighted two-dimensional TV to reconstruct two-dimensional images. This extension requires scrutinizing TV discretization and edge detection approaches. We analyze both the isotropic and anisotropic TV discretizations and describe corresponding edge detection approaches in this subsection.

For demonstration brevity, our presentation is based on incomplete spectral measurements, namely, sampling operator  $\Psi = F_p$ , which is a normalized partial Fourier ensemble defined as  $F_p := PF$ , where  $P \in \mathbb{R}^{m \times n}$  is a down-sampling matrix. The results, however, apply to general images and sampling modalities. In addition to weighted TV, one can also use an extra  $\ell_1$  norm of sparsifying transform  $\Phi$  such as wavelets or curvelets to assist the reconstruction (cf. (2.2)).

**2.2.1. Isotropic TV.** For a two-dimensional image  $u$ , one isotropic discretization of TV using forward difference is defined as

$$TV^{iso}(u) := \sum_{i,j} \|D_{i,j}u\|_2 = \sum_{i,j} \sqrt{(u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2}.$$

With isotropic TV, we use the following iterative scheme to recover  $\bar{u}$ .

**Algorithm: Isotropic EdgeCS.**

*Input:*  $\Phi, b$ .

1. Iteration number  $k \leftarrow 1$ ; weights  $g_{i,j} \leftarrow 1 \forall i, j$ .
2. While the stopping condition is not met, iterate
  - (a) Subproblem:  $u^{(k)} \leftarrow$  solve weighted TV problem

$$(2.2) \quad \min_u \mu \sum_{i,j} g_{ij} \|D_{ij}u\|_2 + \lambda \|\Phi u\|_1 + \frac{1}{2} \|F_p u - b\|_2^2.$$

- (b) Apply edge detection on  $u^{(k)}$  to obtain pixel set  $E^k$  consisting of pixels on edges.
- (c) Weight update for (2.2):  $g_{ij} \leftarrow 0 \forall (i, j) \in E^k$ ;  $g_{ij} \leftarrow 1 \forall (i, j) \notin E^k$ .
- (d)  $k \leftarrow k + 1$ .

In (2.2),  $\mu$  and  $\lambda$  are two weight parameters that balance TV and wavelet sparsity. The motivation of assigning zero TV weight ( $g_{ij} \leftarrow 0$ ) to pixels on edges is to free them from the TV minimization and thus encourage the edges to form in the recovery. Clearly, this assumes that those edges indeed exist and are correctly detected; otherwise, the zero weight would encourage *false* edges in the recovery. Therefore, faithful edge detection is of critical importance.

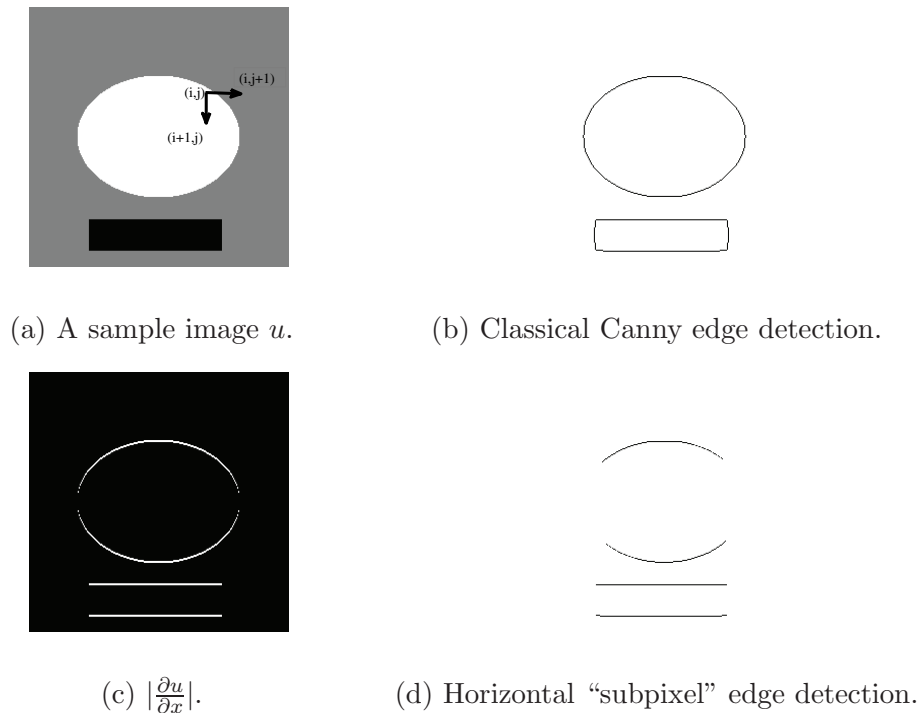
**2.2.2. Anisotropic TV.** In the isotropic TV-based model (2.2), the edge set  $E$  contains the pixels on edges, and these pixels are defined on grids. If  $(i, j) \in E$ , then  $g_{i,j}$  is set to 0, which frees the whole term  $\|D_{i,j}u\|_2$  from minimization. As a result, both  $u_{i+1,j}$  and  $u_{i,j+1}$  are now detached from  $u_{i,j}$ , and hence their differences in the recovery are likely to be larger. This is fine if both pairs of pixels,  $(i, j) \sim (i + 1, j)$  and  $(i, j) \sim (i, j + 1)$ , are cut by edges. However, this is not common. More often seen is just one of them cut by an edge and not both. An example is given in Figure 2, where there is an edge only between  $(i, j)$  and  $(i, j + 1)$ . In such a case, freeing just one pair from minimization is more desirable. Nevertheless, the two pairs are nonlinearly coupled by the 2-norm. A compromise is to assign, instead of weight 0, a small yet nonzero weight (e.g., 0.3) to  $g_{ij}$ . This works better. Columns 2 and 3 of Figure 7 demonstrate its advantage over the 0 weight.

Yet another possible approach is to separately weigh the two pairs inside the 2-norm, i.e., applying a weighted 2-norm in the form of  $\sqrt{g_a \cdot a^2 + g_b \cdot b^2}$  with nonnegative weights  $g_a$  and  $g_b$ . However, this leads to complications begging further investigation.

We found that it is rather easy to use anisotropic TV discretization and assign a weight to each  $|u(i, j) - u(i', j')|$ . More specifically, we follow the same scheme described in section 2.2.1 but substitute the weighted isotropic TV by the following weighted anisotropic TV:

$$\sum_{\alpha} g_{\alpha} |D_{\alpha}u| := \sum g_{(i,j) \sim (l,m)} |u_{i,j} - u_{l,m}|,$$

where the sum is taken over all pairs of neighbors  $(i, j)$  and  $(l, m)$ , which are denoted by  $\alpha$ . The initial values of  $g_{(i,j) \sim (l,m)}$  are determined by the anisotropic TV discretization, as shown in Figure 3. There are three major types of two-dimensional images. More neighbors lead to better approximation to the original TV. If an edge between  $(i, j)$  and  $(l, m)$  is detected,  $g_{(i,j) \sim (l,m)}$  is set to 0. This algorithm is described as follows.



**Figure 2.** Comparison of regular edge detection (b) and subpixel one (d). In (b), edges are detected based on normal of the gradients, and  $g_{i,j}$  is 0 (with black intensity) when  $(i, j)$  is on edge. In (d), subpixel edges are detected only on partial derivatives. For instance, using partial derivatives along the  $x$ -axis, horizontal edges are detected.  $g_{(i,j)\sim(i+1,j)}$  is 0 (black) when there is an edge between  $(i, j)$  and  $(i + 1, j)$ .

#### Algorithm: Anisotropic EdgeCS.

Input:  $\Psi, b$ .

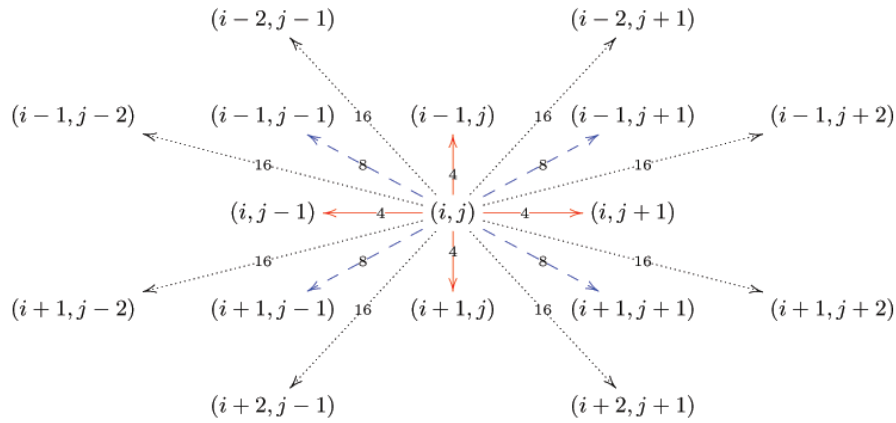
1. Iteration number  $k \leftarrow 1$ ; initialize  $g_{(i,j)\sim(l,m)}$  according to Figure 3.
2. While the stopping condition is not met, do
  - (a) Subproblem:  $u^{(k)} \leftarrow$  solve weighted TV minimization

$$(2.3) \quad \min_u \mu \sum g_{(i,j)\sim(l,m)} |u_{i,j} - u_{l,m}| + \lambda \|\Phi u\|_1 + \frac{1}{2} \|F_p u - b\|_2^2.$$

- (b) Apply subpixel edge detection on  $u^{(k)}$ .
- (c) Weight update:  $g_{(i,j)\sim(l,m)} \leftarrow 0$  if there is an edge between  $(i, j)$  and  $(l, m)$ .
- (d)  $k \leftarrow k + 1$ .

Anisotropic TV permits subpixel (i.e., between-pixel) edge detection, which we explain in the next section. With the original and subpixel (see the discussion of edge detection in section 3) versions of the Canny edge detector, we found that a weighted anisotropic TV, even using the simplest 4-neighbor version, performs much better than weighted isotropic TV.

**2.3. Complex-valued images.** Complex-valued images are widely used in medical imaging. For example, both magnetic resonance (MR) measurements and images are complex-valued, and they provide phase (argument) information that could be used to discriminate



**Figure 3.** A graph representing three different anisotropic TV discretizations based on the 4-, 8-, and 16-neighborhoods of  $(i, j)$ . These neighborhoods consist of those nodes connected to  $(i, j)$  by the solid, dashed, and dotted arcs, respectively. The original weights are as follows [3]: for the 4-neighbor, all weights equal  $\pi/4$ ; for the 8-neighbor type, the nearest 4 neighbors have weights  $\pi/8$ , and the next 4 neighbors have  $\sqrt{2}\pi/16$ ; for the 16-neighbor type, the nearest 4 neighbors have weights  $\frac{1}{2} \tan^{-1}(\frac{1}{2})$ , the next 4 neighbors have  $\frac{\sqrt{2}}{4}(\frac{\pi}{4} - \tan^{-1}(\frac{1}{2}))$ , and the last 8 neighbors have  $\frac{\sqrt{5}}{80}\pi$ .

between fat and water. Since CS imaging potentially has wide applications in medical imaging, we extend EdgeCS to complex-valued images and, furthermore, exploit the fact that the real and imaginary counterparts of an image share edge locations.

Existing work addressing complex-valued image recovery in an MRI context includes [22, 21, 36, 46]. In [22, 21],  $\ell_1$  sparsity on both the real and imaginary components are assumed, and homotopic  $\ell_0$  sparsity is used in [36]. More recently, separate regularity is applied on the magnitude ( $\ell_1$ ) and the phase (smooth) in [46]. We can apply both isotropic and anisotropic EdgeCS by minimizing the sum of the TVs of the real and imaginary components. As an alternative, below we apply joint sparsity between the real and imaginary counterparts under some reasonable assumptions in MRI.

Let  $u_r, u_i, u_m$ , and  $\theta$  be the real part, imaginary part, magnitude, and phase of  $u$ , respectively. In MRI, phase variations occur at the tissue interfaces where magnetic susceptibility changes. Under the assumption that the susceptibility is homogeneous (constant) in one tissue and it changes to another constant in another tissue, the phase field is piecewise constant. For simplicity, we examine a one-dimensional signal  $u$ . From  $u_r(x) = u_m \cos(\theta(x))$ ,  $u_i = u_m \sin(\theta(x))$ , it follows that

$$(2.4) \quad \begin{aligned} u'_r(x) &= u'_m(x) \cos(\theta(x)) - u_m(x) \sin(\theta(x))\theta'(x), \\ u'_i(x) &= u'_m(x) \sin(\theta(x)) + u_m(x) \cos(\theta(x))\theta'(x). \end{aligned}$$

In a domain  $\Omega$  where  $\theta$  is constant,  $\theta'(x) = 0$ , which implies

$$(2.5) \quad u'_r(x) = u'_m(x) \cos(\theta(x)), \quad u'_i(x) = u'_m(x) \sin(\theta(x)), \quad x \in \Omega,$$

i.e.,  $u'_r, u'_i, u'_m$  are multiples of each other. Therefore, the real and imaginary parts share edges in  $\Omega$ .



The following algorithm recovers a complex-valued image with the help of joint edge detection in  $(u_r^{(k)}, u_i^{(k)})$ .

**Algorithm: Complex EdgeCS.**

*Input:*  $\Psi, b$ .

1. Iteration number  $k \leftarrow 1$ ; initialize weights  $g_{(i,j) \sim (l,m)}$  according to Figure 3.
2. While the stopping condition is not met, iterate
  - (a) Subproblem:  $(u_r^{(k)}, u_i^{(k)}) \leftarrow$  solve the weighted TV problem

$$(2.6) \quad \min_{(u_r, u_i)} \mu \sum_{g_{(i,j) \sim (l,m)}} (\|D_{ij}u_r\|_1 + \|D_{ij}u_i\|_1) + \lambda \|\Phi u\|_1 + \frac{1}{2} \|F_p u - b\|_2^2.$$

- (b) Apply subpixel Canny edge detection on  $(u_r^{(k)}, u_i^{(k)})$  jointly.
- (c) Weight update:  $g_{(i,j) \sim (l,m)} \leftarrow 0$  if there is an edge between  $(i, j)$  and  $(l, m)$ .
- (d)  $k \leftarrow k + 1$ .

**2.4. Difference from Bregman regularization.** Bregman regularization [26, 43] applied to TV is related to, but different from, directly reweighing the TV terms. Taking the one-dimensional signal  $u$  as an example, in TV-induced Bregman regularization, the subgradient  $p_i$  of the  $i$ th TV term,  $|(Du)_i|$ , is given at the end of each iteration and used to define the  $i$ th term of the Bregman regularizer,  $|(Du)_i| - \langle p_i, (Du)_i \rangle$ , in the next iteration.  $p_i$  indicates signed/directional edges at  $i$ .  $p_i = 1$  indicates a positive jump at  $i$ , and  $p_i = -1$  indicates a negative jump. If  $p_i = 1$ , the new regularizer term  $|(Du)_i| - \langle p_i, (Du)_i \rangle$  applies 0 penalty to  $(Du)_i \geq 0$ , and a penalty of  $2|(Du)_i|$  to  $(Du)_i < 0$ . The latter is double penalty for sign mismatching between  $p_i$  and the next  $(Du)_i$ . Such an unequal penalty is a double-edged sword since the previous solution may have errors and artifacts and  $p_i$  may be incorrect. If  $p_i$  is incorrect, a low or 0-weighted penalty on  $|(Du)_i|$  is better. One benefit of Bregman is its simplicity as  $p$  is given automatically whereas EdgeCS requires edge detection for better quality. In our simulations below, EdgeCS and Bregman results are given in Figures 9 and 11 for comparison. The image of EdgeCS is sharper and cleaner.

**3. Edge detection.** This section focuses on developing customized edge detectors for EdgeCS, which are different from general edge detectors in various ways as our goal is not merely the edges but improved image reconstruction. First, while general edge detection recovers step edges (discontinuities in intensity), dirac edges (momentary changes of intensity), and fractal edges (which are caused by noise present in the image), as well as other edges, only the step edges are needed by EdgeCS for setting the TV weights. Second, as reconstructed from undersampled measurements, the images subject to edge detection in EdgeCS usually contain more artifacts and errors than normal images. An example is the image at the bottom left in Figure 7, which is a reconstruction of the first EdgeCS iteration. Finally, while general edge detectors produce edges compromising accuracy, completeness, and in some cases, smoothness, EdgeCS needs detected edges to be accurate in location while allowing missing edges. This point is revealed in the analysis for one-dimensional jump detection. In all, we shall develop edge detectors for EdgeCS that detect *reliable* edges from the intermediate images of the iterative scheme EdgeCS.

In what follows, we discuss some existing edge detectors and explain how to adapt them for our needs. Their performances on EdgeCS are also compared.

### 3.1. Some existing intensity based on pixel edge detectors.

1. *Prefiltering and differentiation based edge detectors.* An important class of existing edge detectors is based on prefiltering, followed by intensity differentiation and thresholding. Work in this vein includes [6, 9, 18, 20, 24, 19]. This class of edge detectors typically takes three steps.

*Step 1.* Removing noise by, for example, convolving with a low-pass filter, e.g., Gaussian filter

$$G_\sigma = \frac{e^{-(x^2+y^2)/2\sigma^2}}{2\pi\sigma^2} \text{ or a median filter.}$$

*Step 2.* Approximating the partial derivatives along  $x$ - and  $y$ -directions through, for instance, convolving with two discrete differentiation kernels

$$k_h = \frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad k_v = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix},$$

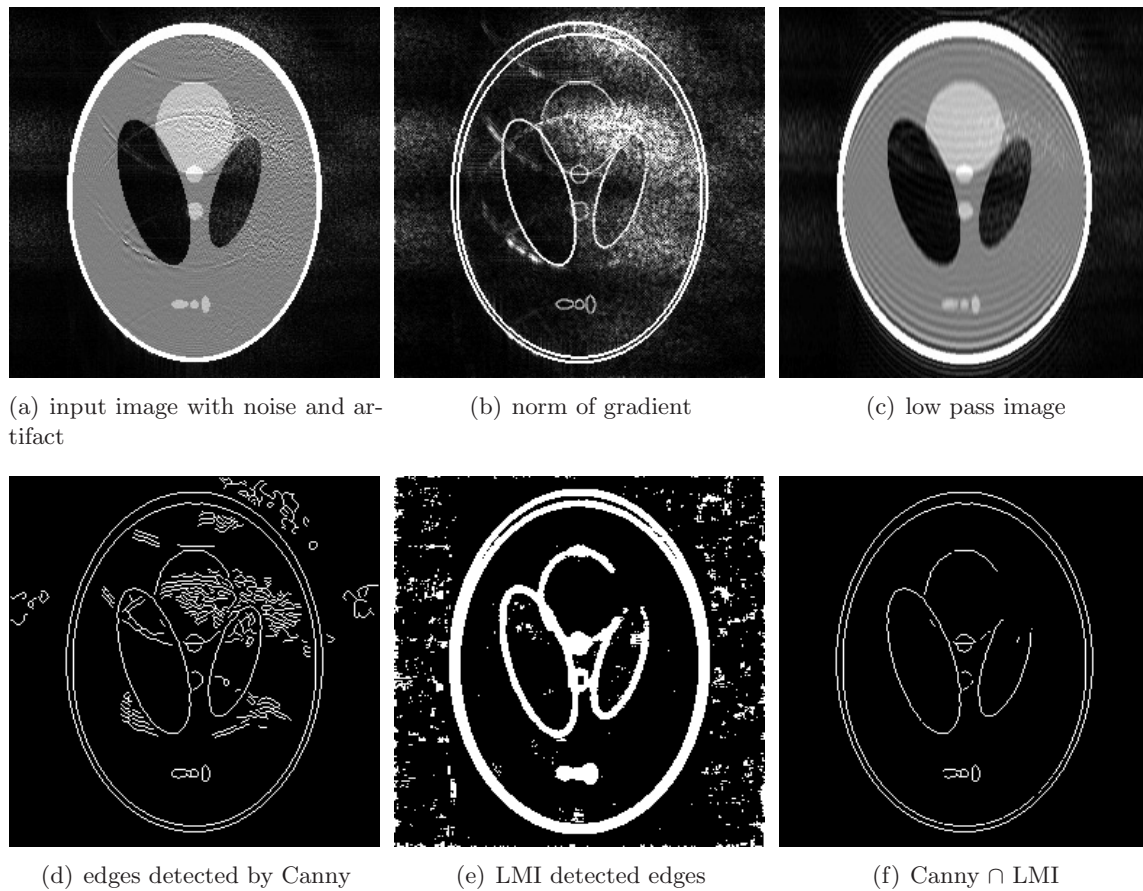
respectively. Various kernels have been developed for this purpose with different accuracies along different directions; see [47, 44] for details.

*Step 3.* Locating edges through thresholding the norm of the gradient magnitude. The standard thresholding treats pixels with gradient magnitude greater than one threshold as edges. Hysteresis thresholding uses two threshold values. Any pixel above the upper threshold is characterized as an edge, and so are those pixels that are in the neighborhood of an edge pixel and with gradients higher than the lower threshold. Hysteresis thresholding leads to connected edges.

Besides the above standard steps, postprocesses such as nonmaximum suppression and smoothing are sometimes conducted to further reduce false positives due to noise. Moreover, instead of maxima of the gradient, zeros of the Laplacian can also be used to locate edges.

The Canny edge detector [6] (`edge(u, 'canny')` in MATLAB) is one of the popular detectors in this class that use hysteresis thresholding on gradients. It is robust to low-level noise, but when the noise/artifact is excessive, it picks up false edges, for it relies on gradient values, which are sensitive to artifacts and noise. In general, it is difficult to tell edges from artifact-induced steps based solely on gradient values.

2. *Local mutual information enhanced edge detector.* To enhance the robustness of the above class of edge detectors to noise and artifacts, one can combine them with local mutual information (LMI) in a method introduced in [16]. Starting with an image  $u$ , it yields an image  $v$  by applying a low-pass filter on  $u$  to remove high-frequency artifacts and noise. At each location  $x$ , it treats  $u(x)$  and  $v(x)$  as random variables and computes the LMI based on their joint and marginal distributions estimated from intensity patches  $\{u(y)\}_{y \in NB_r(x)}$  and  $\{v(y)\}_{y \in NB_r(x)}$ , where  $NB_r(x)$  denotes an  $r \times r$  patch centered at  $x$ . LMI provides a nonnegative measure of the local dependence between  $u$  and  $v$  at every location. Intuitively it measures how many common characteristics are shared between  $u$  and  $v$  pixel wisely. It is zero only when  $u(x)$  and  $v(x)$  appear to be independent, i.e., when their patches near  $x$  are quite different. The more similar the patches are, the higher the LMI will be. Therefore, since strong edges show up in both  $u$  and  $v$ , LMI values are high there. While in locations with



**Figure 4.** Illustration of local mutual information (LMI) based edge detection.

high level noise and artifacts,  $u$  and  $v$  have different patches, and LMI values are lower. LMI is thus able to separate real edges from noise/artifacts to some extent. However, it returns thick boundaries due to its dependence on the probability density function (p.d.f.) (see Figure 4(e)). Since Canny returns thin boundaries and is better on the image background (see Figure 4(d)), intersecting Canny and LMI, written as  $\text{Canny} \cap \text{LMI}$ , leads to more reliable edges (set  $E$  in EdgeCS), as shown in (f).

The disadvantages of LMI are its slow computation of p.d.f. and its performance depending on the low-pass filter used to create  $v$ . The wavelet method below is both accurate and fast.

3. *Wavelet-based edge detectors.* A piece of image is regular if it can be approximated by a polynomial. As wavelets can detect changes in regularity at different scales, the wavelet method is able to detect edges [23] and to even characterize their types. Below, we review the notion of Lipschitz regularity and the dyadic wavelet transform.

**Definition.** Let  $0 \leq \alpha \leq 1$ . A function  $f(x)$  is uniformly Lipschitz  $\alpha$  over an interval  $(a, b)$  if there exists a constant  $K$  such that for any  $x_0, x_1 \in (a, b)$ ,  $|f(x_0) - f(x_1)| \leq K|x_0 - x_1|^\alpha$ .

**Definition.** Let  $\theta(x)$  be a smoothing function that satisfies  $\int_{-\infty}^{\infty} \theta(x) dx = 1$ ,  $\lim_{x \rightarrow -\infty} \theta(x) = 0$ , and  $\lim_{x \rightarrow \infty} \theta(x) = 0$ . Let  $\omega(x) := \theta'(x)$ , which is a wavelet itself since  $\int_{-\infty}^{\infty} \omega'(x) dx = 0$ . The

wavelet transform of  $f$  at scale  $s$  is defined as  $W_s f(x) := f * \omega_s(x)$ , where  $\omega_s(x) = \frac{\omega(x/s)}{s}$  is the scaled wavelet. For dyadic wavelet transforms, the scale  $s$  is chosen as  $s = 2^j$ ,  $j = 1, 2, \dots$

Lipschitz regularity is difficult to verify directly and thus not practically useful, but the following theorem relates local Lipschitz regularity with the dyadic wavelet transform.

**Theorem** (see [23]). *Let  $0 < \alpha < 1$ . A function  $f(x)$  is uniformly Lipschitz  $\alpha$  over  $(a, b)$  if and only if there exists a constant  $K > 0$  such that the wavelet transform satisfies  $|W_{2^j} f(x)| \leq K(2^j)^\alpha$  for all  $x \in (a, b)$  and  $j = 1, 2, \dots$*

According to this theorem, if  $\sup_{x \in (a, b)} |W_{2^j} f(x)|$  strictly decreases as  $j$  increases, then  $-1 \leq \alpha < 0$ , and there is an impulse at  $x$ ; if the maximum increases with scale, then  $0 < \alpha \leq 1$ , and the signal/image intensity changes gradually and thus is smooth at  $x$ . When the maximum does not change much across scales,  $\alpha = 0$ , and there is a step edge at  $x$ . EdgeCS needs step edges, so the set  $E$  includes locations with  $\alpha = 0$ .

4. *Other methods.* Other edge detectors include segmentation [25, 2, 7], morphological gradient [30, 27, 28], and fractal geometry [45, 35], as well as high order and variable order TV-based [33] methods. We have tested a subset of them with EdgeCS, but the results are not competitive. Due to space limitations, we leave them out of our discussion.

**3.2. On-pixel edge detector comparisons.** We conduct two sets of comparisons. One compares the three edge detectors Canny, Canny  $\cap$  LMI, and the wavelet method on two intermediate reconstructions of RecPF [42]. The other compares them in the isotropic EdgeCS framework.



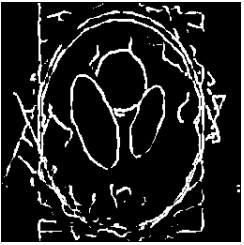
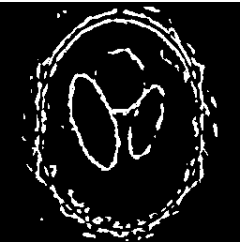
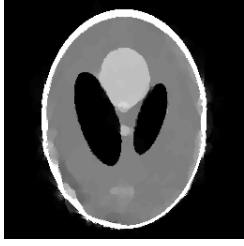
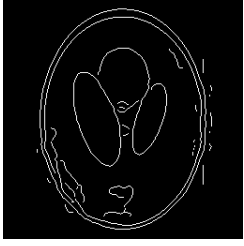
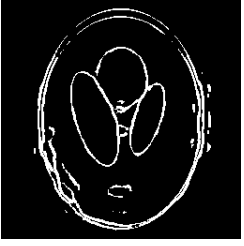
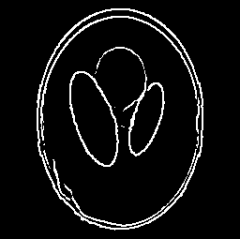
RecPF iteratively recovers an image from its incomplete Fourier samples based on solving

$$(3.1) \quad \min_u \mu \text{TV}(u) + \lambda \|\Phi u\|_1 + \frac{1}{2} \|F_p u - b\|_2^2,$$

where  $\text{TV}(u)$  can be either isotropic or anisotropic.

For the  $256 \times 256$  Shepp–Logan phantom, a set of incomplete Fourier measurements were collected on eight spectral lines (3.98% samples). The three edge detectors were applied to images  $u$  at the end of RecPF iterations 20 and 100. The edge detection results are given in Figure 5. At iteration 20,  $u$  had severe artifacts, and some of them were falsely recognized as edges by Canny and Canny  $\cap$  LMI. The wavelet detector was less affected. At iteration 100,  $u$  had fewer artifacts. As a result, all three detectors produced better edges than previously. In particular, the wavelet edges were nearly exact and contained fewer false positives. Since EdgeCS is sensitive to false edges, the wavelet detector appears to be the best choice. Furthermore, the wavelet detector took just one fifth of the time needed by Canny  $\cap$  LMI.

We plugged each of the three edge detectors into isotropic EdgeCS and compared their performance. From the same measurements above, intermediate reconstructions and detected edges at iterations 50, 300, and 2000 of EdgeCS are given in Figure 6. One can see that even after 2000 iterations EdgeCS with Canny still failed to reconstruct an accurate image, while EdgeCS with either Canny  $\cap$  LMI or the wavelet method succeeded after only 300 steps. EdgeCS with Canny  $\cap$  LMI eventually returned slightly better results than with the wavelet method. However, the former one was much slower. Considering both quality and speed, we found that the wavelet edge method detector is the best fit for isotropic EdgeCS, and it was used in the simulation presented in section 5.

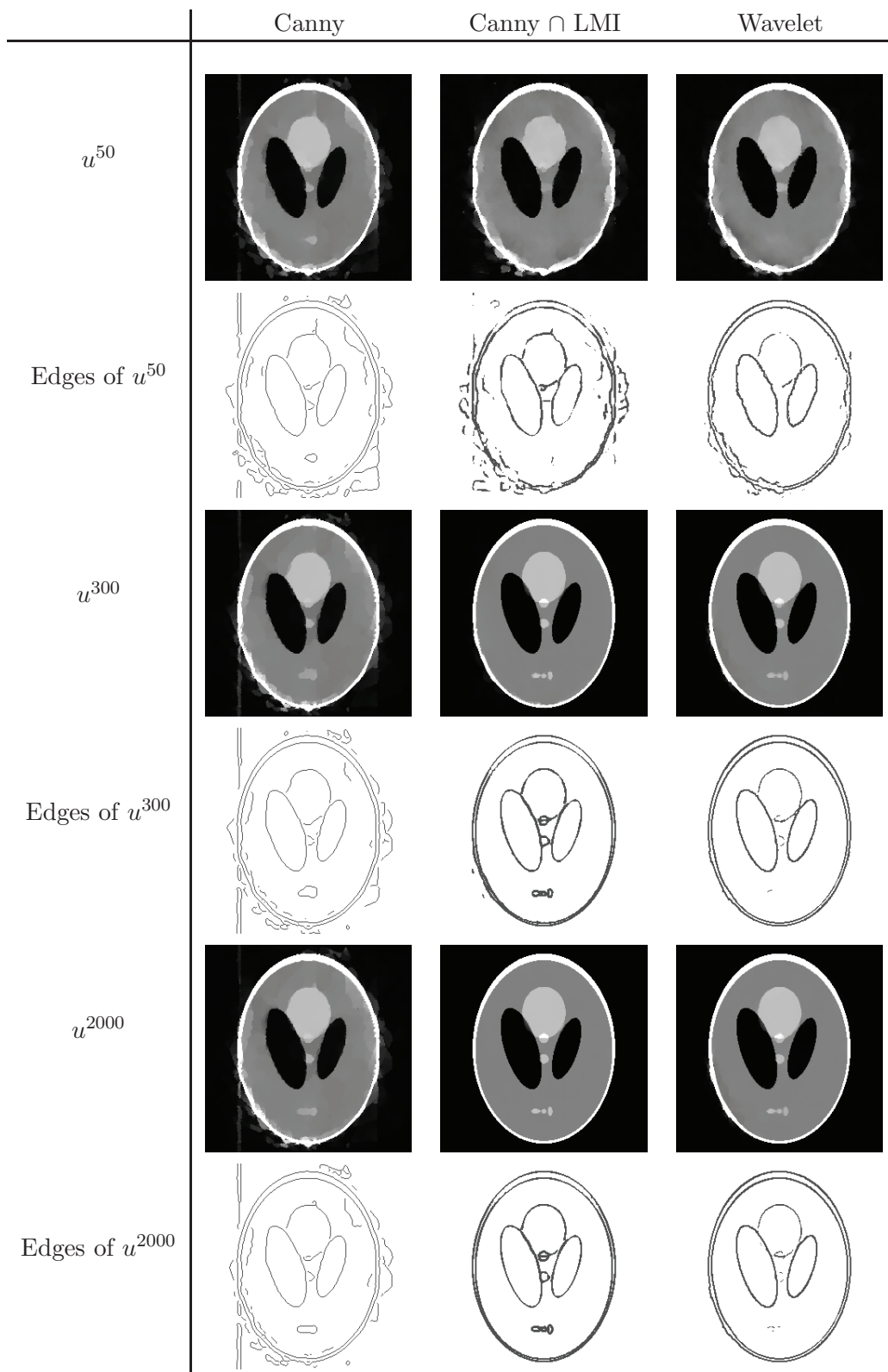
Itr. $k$	$u^{(k)}$	Canny	Canny $\cap$ LMI	Wavelets
20				
100				
Average Seconds:		0.05	1.98	0.37

**Figure 5.** Comparison of three edge detectors on two intermediate RecPF reconstructions from spectral measurements taken on 8 radial lines (3.98%).

**3.3. Subpixel edge detection.** The anisotropic EdgeCS in subsection 2.2.2 requires subpixel edges. We have modified Canny for subpixel edge detection, but it is difficult to do so on Canny  $\cap$  LMI and the wavelet edge detector. The original Canny applies hysteresis thresholding on gradient magnitudes  $\|D_{ij}u\|_2$ . For subpixel edges, we let hysteresis thresholding be applied to directional gradients  $D_\alpha u$ , e.g.,  $|u(i, j) - u(i + 1, j)|$  and  $|u(i, j) - u(i, j + 1)|$ . The Canny subpixel edge detector was used with both anisotropic and complex EdgeCS in the simulation presented in section 5.

**4. Algorithms.** This section provides the implementation details of EdgeCS. Our implementation is based on iterative edge detection and the recent solvers RecPF [42] with split Bregman [15], which are applied to (2.3). RecPF and split Bregman can both be derived from the well-known alternating direction method of multipliers (ADMM) (cf. [14]), and they are very efficient because they break a nonsmooth multiterm optimization problem (3.1) into subproblems with closed-form solutions. This advantage for TV regularization problems was first discovered in [38] for image denoising and deblurring and was generalized to multichannel problems in [40], the TV- $L^1$  model in [41], and TV-based CS in [42] and [15]. Since we need weighted TV, which is not covered in the above work, we briefly describe the algorithm for (2.3) with weighted anisotropic TV and discuss its performance. It extends to isotropic and complex versions easily.

Consider an  $m \times n$  image. By introducing  $H(u, b) = \frac{1}{2}\|F_p u - b\|_2^2$  and auxiliary variables  $z \in \mathbb{C}^{m \times n}$  and  $w = [w_1, \dots, w_{m \times n}]$ , where each  $w_i \in \mathbb{C}^2$ , we can rewrite (2.3) as the equivalent



**Figure 6.** Comparison of three on-pixel edge detectors on three intermediate isotropic EdgeCS reconstructions from spectral measurements taken on 8 radial lines (3.98%).

constrained problem

$$(4.1) \quad \min_{u,w,z} \mu \sum_{\alpha} g_{\alpha} |w_{\alpha}| + \lambda \|z\|_1 + H(u, b) \quad \text{s.t. } z = \Phi u, \quad w_{\alpha} = D_{\alpha} u \quad \forall \alpha,$$

where  $\alpha$  stands for a pair of neighbor pixels and  $D_{\alpha} u$  is their difference. To simplify notation, we introduce

$$J_g(w) := \mu \sum_{\alpha} g_{\alpha} |w_{\alpha}|, \quad K(z) := \lambda \|z\|_1.$$

The augmented Lagrangian of (4.1) is

$$(4.2) \quad \mathcal{L}(u, w, z) := J_g(w) + \frac{\mu\beta}{2} \|w - Du - d_w\|_2^2 + K(z) + \frac{\lambda\beta}{2} \|z - \Phi u - d_z\|_2^2 + H(u, b),$$

where  $d_w$  and  $d_z$  are Lagrange multipliers of proper sizes. The problems of minimizing  $\mathcal{L}(u, w, z)$  with respect to each of  $u$ ,  $w$ , and  $z$  while fixing the rest are called the  $u$ -,  $w$ -, and  $z$ -subproblems:

$$(4.3) \quad u\text{-subproblem: } \min_u \frac{\mu\beta}{2} \|w - Du - d_w\|_2^2 + \frac{\lambda\beta}{2} \|z - \Phi u - d_z\|_2^2 + H(u, b),$$

$$(4.4) \quad w\text{-subproblem: } \min_w J_g(w) + \frac{\mu\beta}{2} \|w - Du - d_w\|_2^2,$$

$$(4.5) \quad z\text{-subproblem: } \min_z K(z) + \frac{\lambda\beta}{2} \|z - \Phi u - d_z\|_2^2.$$

The ADMM applied to (4.1) or, equivalently, (2.3) is as follows.

**Algorithm 1.** Given  $b$ ,  $\Phi$ ,  $F_p$ ,  $\lambda$ ,  $\mu$ :

1. Normalize input  $b$  and parameters  $\lambda$  and  $\mu$ .
2. Set  $\beta$  and  $\gamma$ .
3.  $k \leftarrow 0$ ,  $u^{(k)} \leftarrow \mathbf{0}$ ,  $d_w^{(k)} \leftarrow \mathbf{0}$ ,  $d_z^{(k)} \leftarrow \mathbf{0}$ .
4. while not converged do
5.  $k \leftarrow k + 1$ ,
6.  $w^{(k)} \leftarrow$  solve (4.4) for  $u = u^{(k-1)}$  and  $d_w = d_w^{(k-1)}$ ,
7.  $z^{(k)} \leftarrow$  solve (4.5) for  $u = u^{(k-1)}$  and  $d_z = d_z^{(k-1)}$ ,
8.  $u^{(k)} \leftarrow$  solve (4.3) for  $d_w = d_w^{(k-1)}$  and  $d_z = d_z^{(k-1)}$ ,
9.  $d_w^{(k)} \leftarrow d_w^{(k-1)} - \gamma(w^{(k)} - Du^{(k)})$  and  $d_z^{(k)} \leftarrow d_z^{(k-1)} - \gamma(\Phi u^{(k)} - z^{(k)})$ ,
10. End while.
11. Denormalize output  $u^k$ .

The  $w$ -subproblem (4.4) is separable in each  $w_i$ , so it is straightforward to derive the minimizer (cf. [40]):

$$w_i^* = \text{shrink} \left( D_i u + (d_w)_i, \frac{g_i}{\beta} \right), \quad \text{where } \text{shrink}(\mathbf{t}, a) := \max\{0, \|\mathbf{t}\| - a\} \cdot \frac{\mathbf{t}}{\|\mathbf{t}\|}$$

and  $0 \cdot (0/0) = 0$  is assumed.

The  $z$ -subproblem (4.5) is also separable in each  $z_i$ , so the minimizer is given by

$$z_i^* = \text{shrink} \left( (\Phi u)_i + (d_z)_i, \frac{1}{\lambda} \right).$$

The  $u$ -subproblem (4.3) is also surprisingly simple to solve. Since the objective function is convex quadratic, the minimizer  $u^*$  is the solution of the normal equations,

$$(4.6) \quad \left( \lambda\beta I + \mu\beta D^\top D + F_p^* F_p \right) u = \lambda\beta \Phi^*(z - d) + \mu\beta D^\top (w - \mathbf{b}) + F_p^* b,$$

where the first term  $\lambda\beta I$  is obtained from the fact  $\lambda\beta \Phi^* \Phi = \lambda\beta I$  because  $\Phi$  is unitary. Under the periodic boundary conditions for  $u$ , the finite difference operator  $D$  is a block-circulant linear operator, so  $D^\top D$  and thus the entire left-hand side matrix in (4.6) can be diagonalized by the discrete Fourier transform  $F$ ; namely,

$$F \left( \lambda\beta I + \mu\beta D^\top D + F_p^* F_p \right) F^* = \lambda\beta I + \mu\beta \hat{D}^* \hat{D} + P^\top P$$

is diagonal (noticing that  $P^\top P$  is a diagonal 0/1 matrix). One does not need to form the matrix in the computation. Define

$$\hat{d} := \text{diag}(\lambda\beta I + \mu\beta \hat{D}^* \hat{D} + P^\top P).$$

Since  $\hat{d}$  remains constant throughout all iterations, it should be computed at the beginning of the algorithm and used repeatedly. The minimizer of (4.3) is given by

$$u^* = F^* \left( F(\lambda\beta \Phi^*(z - d) + \mu\beta D^\top (w - \mathbf{b}) + F_p^* b) ./ \hat{d} \right),$$

where  $./$  stands for componentwise division. As only two fast Fourier transforms (FFTs) are needed, solving the  $u$ -subproblem (4.3) is simple and fast.

The algorithm is GPU-friendly since besides FFTs, which already have GPU implementations, the computations are local to pixels. A preliminary version has been implemented and tested on nVidia GPUs with promising speedup.

**Parameters and performance.** The parameters  $\beta$  and  $\gamma$  in Algorithm 1 only moderately affect the convergence speed. It is proved in [14] that  $\gamma < (\sqrt{5} + 1)/2$  guarantees global convergence. In our test, we found that  $1 \leq \gamma < (\sqrt{5} + 1)/2$  consistently yields good performance. The other parameter  $\beta$  must be strictly positive but not upper bounded. To make  $\beta$  simple to choose, we normalize the input  $b$  and the regularization parameters  $\lambda$  and  $\mu$  in order to remove the variations due to the image size  $m \times n$ , pixel intensity range, and the number of measurements  $k$  (i.e., the size of  $b$ ). Specifically, both  $\lambda$  and  $\mu$  are multiplied with  $k/\sqrt{mn}$  for the normalization with respect to the image size and  $k$ , and  $b$  is rescaled by dividing the pixel intensity range (e.g., 255 for 8-bit or 65535 for 16-bit). With such normalization, we found that  $\beta$  between 5 and 20 consistently leads to good performance. Alternatively, one can increase  $\beta$  over the iterations; refer to [14] for details. Note that the above normalization also makes  $\lambda$  and  $\mu$  relatively independent of the three factors mentioned above, and they remain dependent on the gradient sparsity of the underlying image and the noise/error level in the measurements.



The algorithm is terminated after a fixed number of iterations. For most problems, 100 and 300 iterations were used for isotropic and anisotropic EdgeCS, respectively. Difficult problems may need more iterations, and in general they lead to higher signal-to-noise ratios (SNRs) and lower errors, especially for problems with very low sampling rates.

## 5. Simulation results.

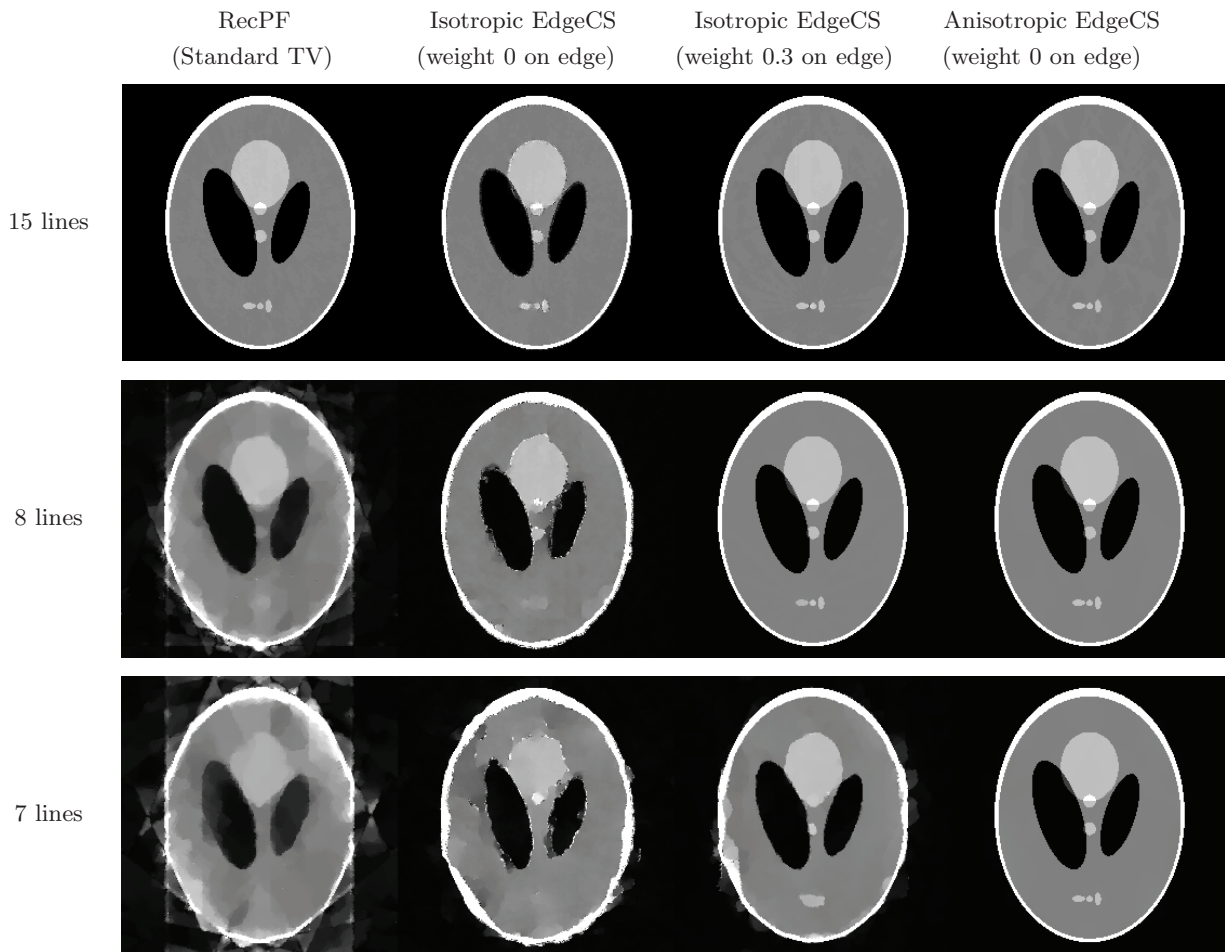
**5.1. Shepp–Logan phantom.** In Figure 7, we compare isotropic EdgeCS with either 0 or 0.3 weights on detected edges, anisotropic EdgeCS, and RecPF on recovering images from 6.44%, 3.98%, and 3.03% (or 15, 8, and 7 radial lines) of spectral measurements of the  $256 \times 256$  Shepp–Logan phantom. As this is the first algorithm that iteratively uses edges to help recover images, there are no similar algorithms to compare it with. We comment that a related paper [34] compares three methods for edge detection from incomplete Fourier measurements, but none of them produces images. Therefore, we compared three EdgeCS algorithms—*isotropic*, *anisotropic*, and *complex*—with RecPF [42], which is based on isotropic TV and does not exploit edge detection. In anisotropic and complex EdgeCS, four neighbors were used for each pixel. These algorithms were compared based on their image reconstructions from significantly undersampled measurements, as well as such measurements with excessive noise.

The parameters  $\mu$  and  $\lambda$  control the overall performance. We fixed  $\lambda$  to 0 in all tests in order to focus on the contributions due to edge detection. Interested readers are referred to [21, 8] for information on a sparsity term ( $\lambda > 0$ ) other than TV. Upon being called, all algorithms scaled the input data and parameter  $\mu$  in the same way in order to normalize the effects due to varying image sizes, pixel intensity ranges, and sample sizes. Specifically, each given  $\mu$  was multiplied by the sample size and divided by the square root of pixel quantity, and  $b$  was divided by the intensity range. The returned images were denormalized by multiplying with the intensity range. All results were obtained with tight parameters to avoid loss of quality due to early stopping. All tests were run under Windows 7 and MATLAB v7.10 (R2010a) on a laptop with an Intel Core 2 Duo CPU at 2.0 GHz and 3 GB of memory.

We normalized the ground truth images to the intensity range  $[0, 1]$ . Their spectral measurements were collected on smooth radial sampling trajectories that are empirically shown to be effective.

Given sufficient measurements (15 radial lines), the four results are visually comparable. In terms of error and SNR, however, anisotropic EdgeCS results are slightly better. From the 8-line measurements, RecPF returned an image with a relative error of 37.15%, but isotropic EdgeCS with .3 weight on edges and anisotropic EdgeCS returned much better images. From the 7-line measurements, anisotropic EdgeCS still achieved an almost exact recovery, while RecPF returned an image with apparent errors and many artifacts, and isotropic EdgeCS was not effective either.

Anisotropic EdgeCS also returned better images than RecPF from noise-contaminated measurements. Figure 8 depicts the images recovered from measurements that are 6.44% of  $k$ -space (or 15 radial lines) added with Gaussian noise of varying variances  $\sigma^2 = 0, 0.01, 0.05$  (recall that the phantom has intensity values between 0 and 1). Corresponding to  $\sigma^2 = 0$  (or no noise), RecPF and anisotropic EdgeCS returned comparable images. Corresponding to  $\sigma^2 = .01$  (or a low level of noise), anisotropic EdgeCS produced an image with sharper edges (observable in the boxed areas). Corresponding to  $\sigma^2 = 0.05$ , anisotropic EdgeCS returned a



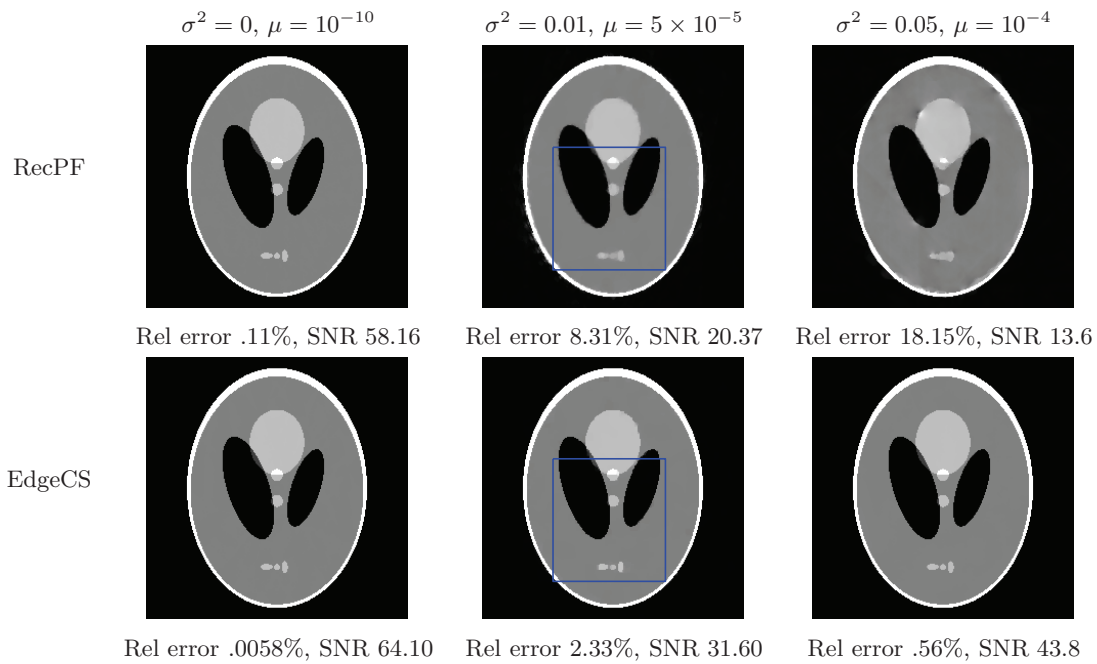
<i>Relative error</i>				
15 lines	.11%	5.00%	.45%	.006%
8 lines	37.17%	30.55%	.74%	.086%
7 lines	51.65%	47.97%	27.90%	1.09%
<i>SNR (dB)</i>				
15 lines	58.16	24.78	45.74	64.10
8 lines	7.4	9.06	41.40	60.10
7 lines	4.5	5.10	9.80	38.00(see caption)

**Figure 7.** Comparison of four reconstruction methods on measurements taken on 15, 8, and 7 radial lines (sample rates 6.44%, 3.98%, and 3.03%, respectively) with no noise added, and parameter  $\mu = 10^{-10}$ . Note: SNR 38dB of anisotropic EdgeCS at 7 lines can be significantly improved if more iterations are allowed.

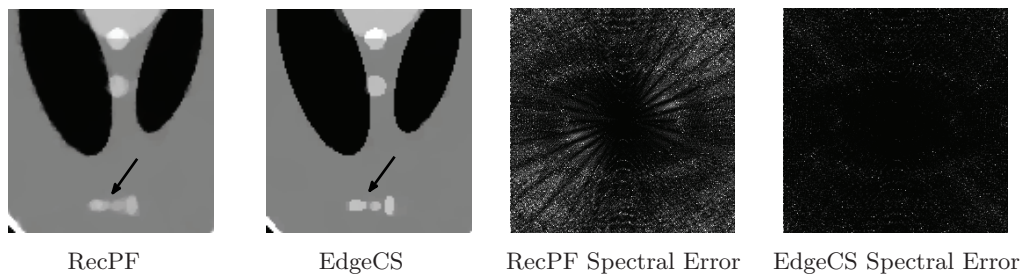
much better image than RecPF, which failed to recover the small shapes.

In Figure 8 we compared the performance of RecPF and anisotropic EdgeCS on Shepp–Logan phantom data with various noise levels. A close-up comparison is given in Figure 9. It can be seen that EdgeCS performs better than RecPF consistently.

We tested how RecPF and EdgeCS perform under different image resolutions. The Shepp–



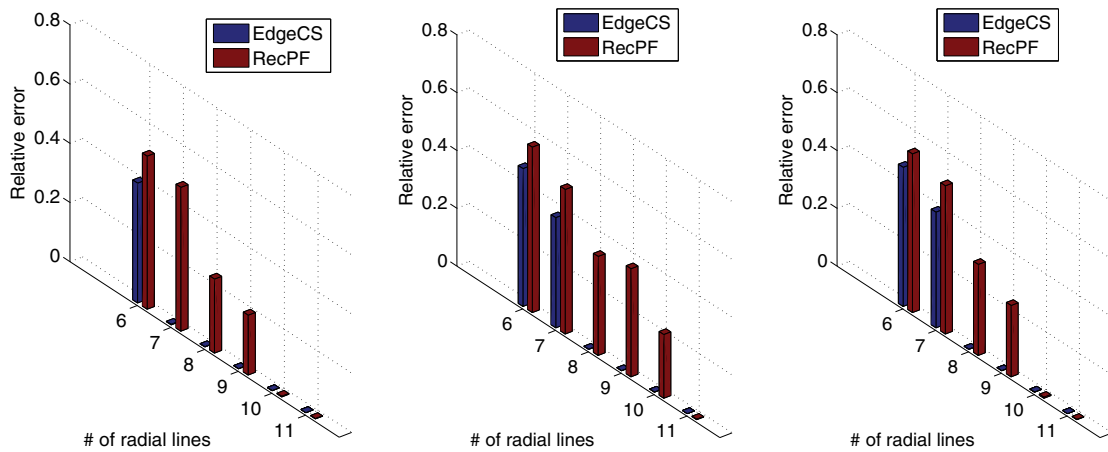
**Figure 8.** Comparisons of RecPF and anisotropic EdgeCS at different noise variances.



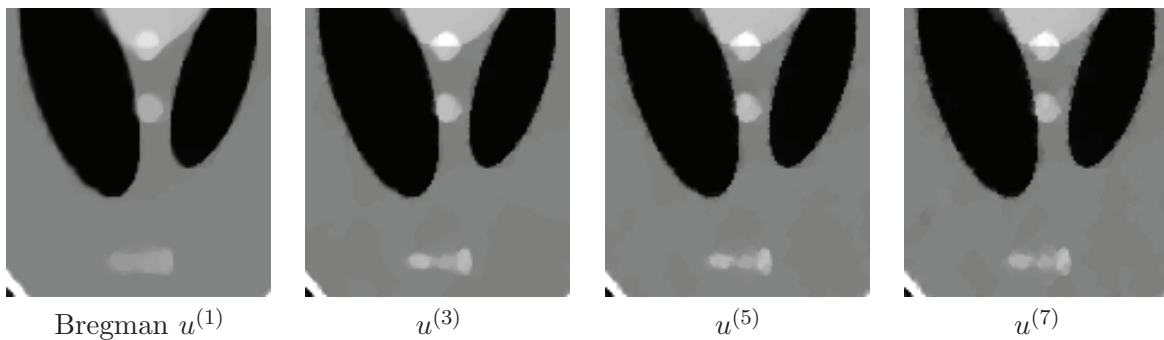
**Figure 9.** Zoom-in comparisons of RecPF and anisotropic EdgeCS results at  $\sigma^2 = 0.01$  and  $\mu = 5 \times 10^{-5}$ . Black arrows point to small geometries that are preserved by anisotropic EdgeCS (middle left) but not by RecPF (left).

Logan phantom images at three different resolutions,  $256 \times 256$ ,  $128 \times 128$ , and  $64 \times 64$ , were sampled over 6, 7,  $\dots$ , 12 radial lines and then recovered by RecPF and anisotropic EdgeCS with the same parameter  $\mu = 10^{-6}$ . The recovery relative errors are compared in Figure 10. The improvement of EdgeCS over RecPF is consistent under the different resolutions.

We also ran Bregman iterations on the noisy measurements. The results are depicted in Figure 11. The first iteration yields an oversmoothed image, and the subsequent iterations become sharper. Since the limit satisfies  $F_p u - b$  and  $b$  contains noise, smears and artifacts appear in the images of further iterations. Figure 11,  $u^{(3)}$ , appears to be best image. However, it is not as sharp and clean as EdgeCS's result.



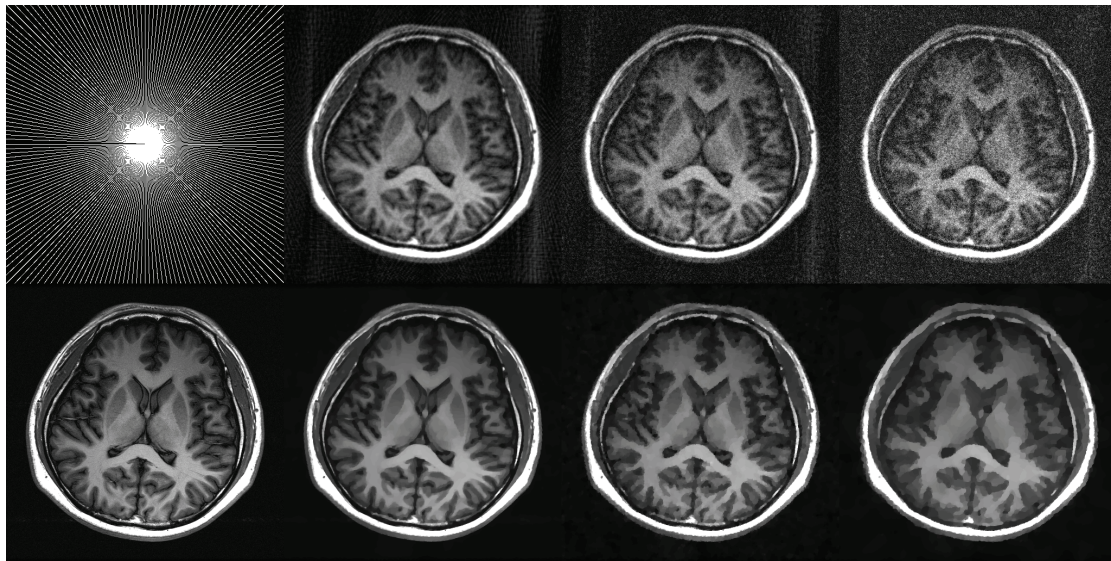
**Figure 10.** Comparison of relative errors of RecPF and EdgeCS recovered Shepp–Logan phantom at different resolutions (left:  $256 \times 256$ ; middle:  $128 \times 128$ ; right:  $64 \times 64$ ). The x-axis is the number of radial lines (6–11); the y-axis is recovery relative error; the left (blue) and right (red) bars correspond to EdgeCS and RecPF, respectively.  $\mu = 10^{-6}$  for all results.



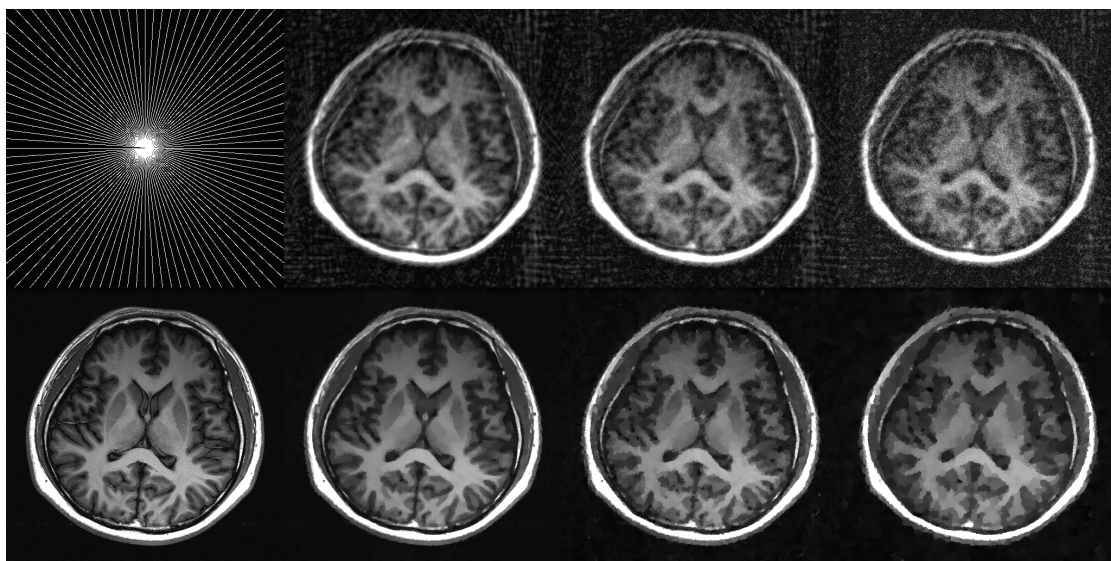
**Figure 11.** Zoom-in comparisons of Bregman iterations 1, 3, 5, and 7.

**5.2. Real-valued MR images.** Since anisotropic EdgeCS appears to perform better than the rest, we continue to test it on reconstructing anatomical MR images. We used a  $512 \times 512$  real-valued fully sampled MR image as the ground truth and simulated different measurements. Figure 12 is based on 100-radial-line sampling (20.87% sampling rate) and noise variances  $\sigma^2 = 0, 0.05$ , and  $0.1$ . Figure 13 is based on 50-radial-line sampling (10.64% sampling rate), and Figure 14 is based on a random sampling trajectory (26.22% sampling rate). It is observed that with a sampling rate as low as 10%, trustworthy results were still recovered even at the noise variance of 0.05. All  $\mu$  values were chosen in between  $10^{-4}$  and  $10^{-3}$ . The entire computation took less than 30 seconds.

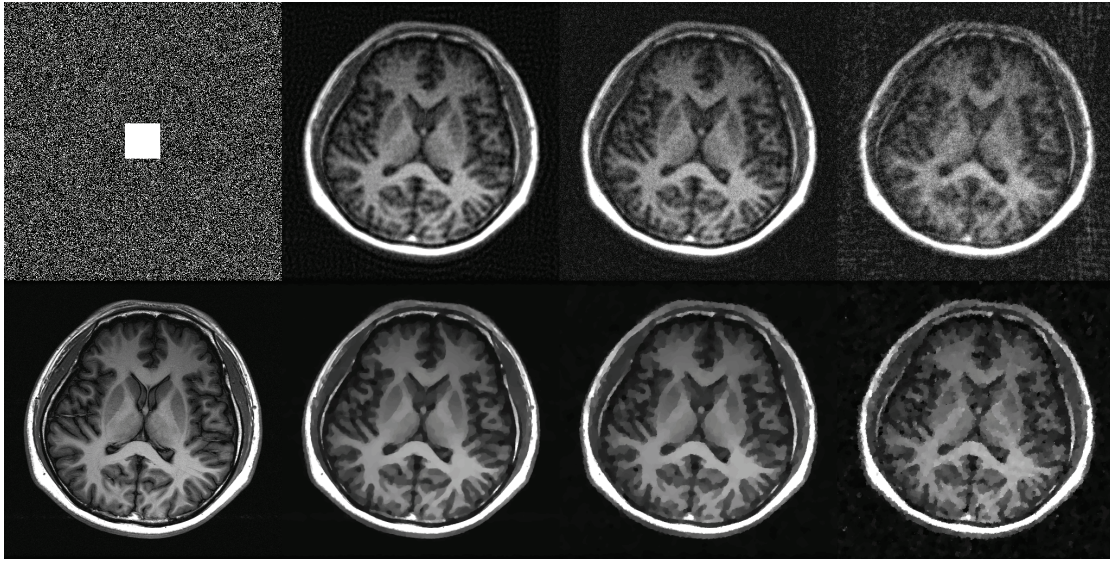
**5.3. Complex-valued MR images.** Motivated by the fact that most of the MR images are complex-valued, we conducted a test on reconstructing complex-valued images. Here, we compare anisotropic EdgeCS to complex EdgeCS. The former treats the real and imaginary parts with one TV term for each with independent weights. The latter uses just one set of



**Figure 12.** Comparison between back projection (top row, last three) and anisotropic EdgeCS (bottom row, last three) on reconstructing a real-world brain slide (bottom row, left) from spectral measurements taken on 100 radial lines (top row, left, 20.87% sampling rate) out of a  $512 \times 512$  domain, added with noise of variances 0, 0.05, and 0.1.



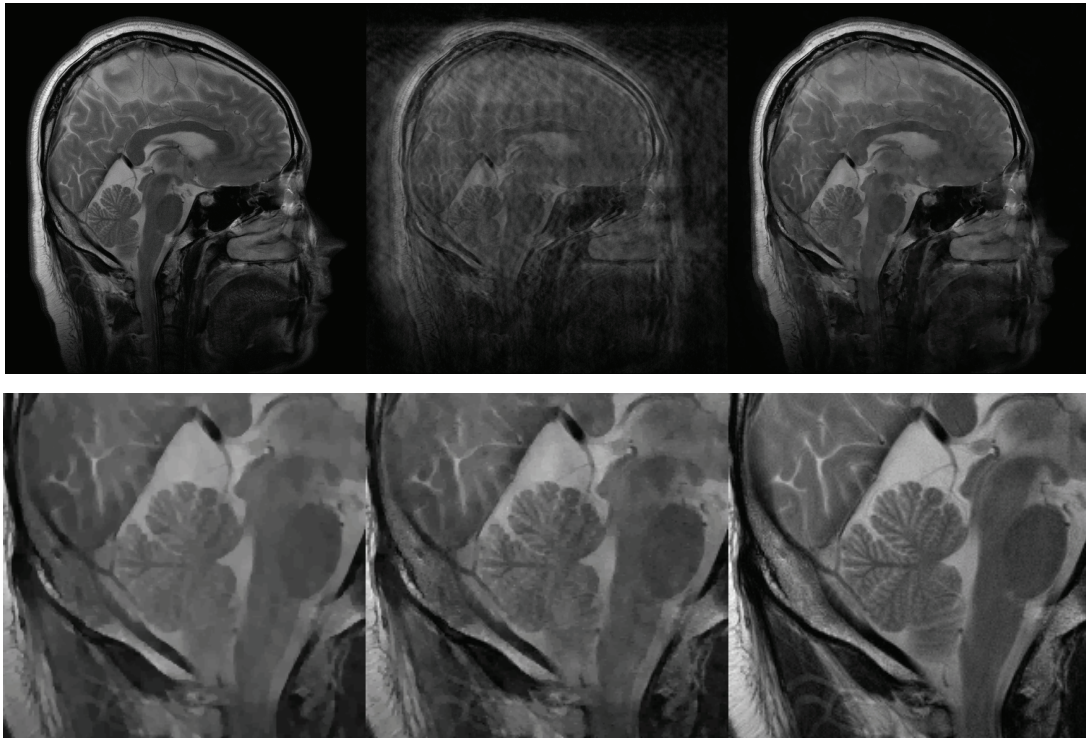
**Figure 13.** Comparison between back projection (top row, last three) and anisotropic EdgeCS (bottom row, last three) on reconstructing a real-world brain slide (bottom row, left) from spectral measurements taken on 50 radial lines (top row, left, 10.64% sampling rate) out of a  $512 \times 512$  domain, added with noise of variances 0, 0.05, and 0.1.



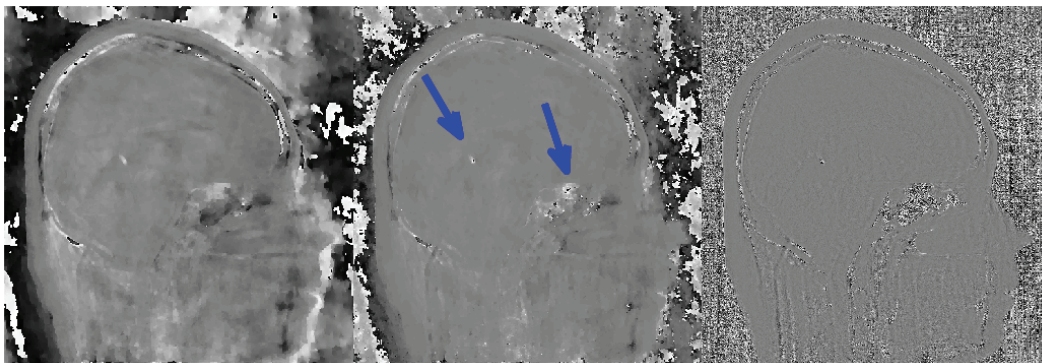
**Figure 14.** Comparison between back projection (top row, last three) and anisotropic EdgeCS (bottom row, last three) on reconstructing a real-world brain slice (bottom row, left) from spectral measurements taken on full-low-plus-random-high frequencies (top row, left, 26.22% sampling rate) out of a  $512 \times 512$  domain, added with noise of variances 0, 0.05, and 0.1.

weights and utilizes the joint edge detection. The results depicted in Figures 15, 16, and 17 indicate that the latter method returned significantly better results. The latter is better than the former in image quality (second row in Figure 15), relative error (21.95% versus 16.36%), SNR (12.5 versus 14.5), phase accuracy (Figure 16), and fine feature preservation (left and middle images of Figure 17). This slice of brain image has rich fine structures that are not so sparse under TV, yet complex EdgeCS largely reconstructed those fine structures.

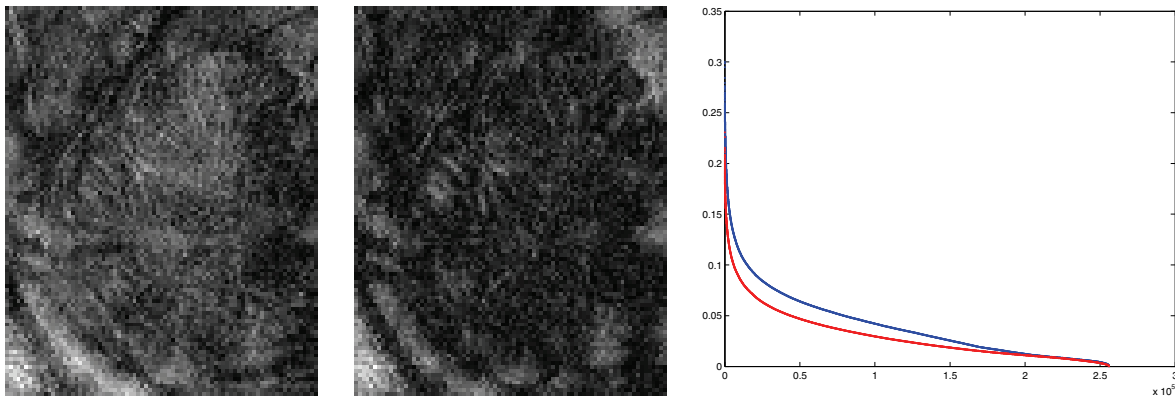
**6. Conclusions.** There is a long history of using edge sparsity and TV minimization for image recovery. Edges are not only sparse but also have other properties, such as connectivity and smoothness, which have been well exploited by various edge detectors. It appears that coupling edge detection and TV minimization lead to more trustworthy reconstructions. We propose in this paper EdgeCS, which requires few measurements and returns images with reduced errors and artifacts than standard TV methods. The edge detection is applied to the intermediate iterates and obtains edges either on pixels or between pixels. The isotropic or anisotropic TV terms corresponding to the detected edges are assigned with low weights. Further investigation shows that the coupling of the two requires the edge detector to be tailored to intermediate image reconstructions. Better performance is observed for complex-valued images by considering joint sparsity between the real and imaginary parts. In spite of getting state-of-the-art recoveries, our methods are rather simple, and we believe that there is much room for further improvement by exploiting the structures of edges, as well as other image geometries, in more effective ways.



**Figure 15.** A complex-valued sagittal brain example (only magnitude images are shown); 200 radial lines out of a  $512 \times 500$  domain; 39.06% sampling rate. Top (left to right): Ground truth, back projection, and anisotropic EdgeCS results. Bottom (left to right): Zoom-in of anisotropic EdgeCS, complex EdgeCS, and ground truth. Relative errors of anisotropic EdgeCS and complex EdgeCS are 21.95% and 16.36%, and SNRs are 12.5 and 14.5, respectively.



**Figure 16.** From left to right: Phase map of anisotropic EdgeCS, complex EdgeCS, and ground truth. Blue arrows point to the areas where complex EdgeCS performs significantly better.



**Figure 17.** Zoom-in comparison of reconstruction errors between anisotropic EdgeCS (left) and complex EdgeCS (middle) on reconstructing a complex-valued brain image. Right plot: Comparison of sorted reconstruction errors (top blue: anisotropic EdgeCS; bottom red: complex EdgeCS).

**Acknowledgment.** The authors thank the referees for their valuable comments, which helped to improve the paper.

#### REFERENCES

- [1] M. AHARON, M. ELAD, AND A. BRUCKSTEIN, *K-SVD: An algorithm for denoising overcomplete dictionaries for sparse representation*, IEEE Trans. Signal Process., 54 (2006), pp. 4311–4322.
- [2] L. AMBROSIO AND V. M. TORTORELLI, *On the approximation of free discontinuity problems* Boll. Un. Mat. Ital. B (7), 6 (1992), pp. 105–123.
- [3] Y. BOYKOV AND V. KOLMOGOROV, *Computing geodesics and minimal surfaces via graph cuts*, in IEEE International Conference on Computer Vision, Vol. 1, IEEE, Washington, DC, 2003, pp. 26–33.
- [4] E. CANDÈS, J. ROMBERG, AND T. TAO, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, 52 (2006), pp. 489–509.
- [5] E. CANDÈS AND T. TAO, *Near optimal signal recovery from random projections: Universal encoding strategies*, IEEE Trans. Inform. Theory, 52 (2006), pp. 5406–5425.
- [6] J. CANNY, *A computational approach to edge detection*, IEEE Trans. Pattern Anal. Mach. Intell., 8 (1986), pp. 679–714.
- [7] T. F. CHAN AND L. A. VESE, *Active contours without edges*, IEEE Trans. Image Process., 10 (2001), pp. 266–277.
- [8] R. COMPTON, S. OSHER, AND L. BOUCHARD, *Hybrid Regularization for MRI Reconstruction with Static Field Inhomogeneity Correction*, UCLA CAM report 11-84, University of California, Los Angeles, CA, 2011.
- [9] R. DERICHE, *Fast algorithms for low-level vision*, IEEE Trans. Pattern Anal. Mach. Intell., 12 (1990), pp. 78–87.
- [10] D. DONOHO, *Compressed sensing*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1289–1306.
- [11] J. DUARTE-CARVAJALINO AND G. SAPIRO, *Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization*, IEEE Trans. Image Process., 18 (2009), pp. 1395–1408.
- [12] M. ELAD, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, New York, 2010.
- [13] M. FRIEDLANDER, H. MANSOUR, R. SAAB, AND O. YILMAZ, *Recovering Compressively Sampled Signals Using Partial Support Information*, preprint, <http://arxiv.org/abs/1010.4612>, 2010.
- [14] R. GLOWINSKI, J. L. LIONS, AND R. TREMOLIERES, *Numerical Analysis of Variational Inequalities*, North-Holland, Amsterdam, 1981.



- [15] T. GOLDSTEIN AND S. OSHER, *The Split Bregman Algorithm for L1 Regularized Problems*, UCLA CAM report 08-29, University of California, Los Angeles, CA, 2008.
- [16] W. GUO AND F. HUANG, *A local mutual information guided denoising technique and its application to self-calibrated partially parallel imaging*, in MICCAI, Part II, Lecture Notes in Comput. Sci. 5242, Springer, New York, 2008, pp. 939–947.
- [17] W. GUO AND W. YIN, *Edges: Edge guided compressive sensing reconstruction*, in Proceedings of SPIE Visual Communication and Image Processing, SPIE, Bellingham, WA, 2010, pp. 77440L-1–77440L-10.
- [18] B. JAHNE, H. SCHARR, AND S. KORGEL, *Principles of filter design*, in Computer Vision and Applications, Volume 2, B. Jahne, H. HauEecker, and P. GeiEJer, eds., Signal Processing and Pattern Recognition, Academic Press, San Diego, CA, 1999, pp. 125–151.
- [19] J. J. KOENDERINK AND A. J. VAN DOOM, *Generic neighborhood operators*, IEEE Trans. Pattern Anal. Mach. Intell., 14 (1992), pp. 597–605.
- [20] S. LANSER AND W. ECKSTEIN, *Eine modification des deriche-verfahrens zur kantendetektion*, in Mustererkennung 1991, B. Radig, ed., Informatik Fachberichte 290, Springer, Berlin, 1991, pp. 151–158.
- [21] M. LUSTIG, D. DONOHO, AND J. PAULY, *Sparse MRI: The application of compressed sensing for rapid MR imaging*, Magnetic Resonance in Medicine, 58 (2007), pp. 1182–1195.
- [22] M. LUSTIG, D. L. DONOHO, J. M. SANTOS, AND J. M. PAULY, *Compressed sensing MRI*, IEEE Signal Process. Magazine, 25 (2008), pp. 72–82.
- [23] S. MALLAT, *Characterization of signals from multiscale edges*, IEEE Trans. Pattern. Anal. Mach. Intell., 14 (1992), pp. 710–732.
- [24] D. MARR AND E. HILDRETH, *Theory of edge detection*, Proc. Roy. Soc. London Ser. B, 207 (1980), pp. 187–217.
- [25] D. MUMFORD AND J. SHAH, *Optimal approximation by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42 (1989), pp. 557–685.
- [26] S. OSHER, M. BURGER, D. GOLDFARB, J. XU, AND W. YIN, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4 (2005), pp. 460–489.
- [27] I. PITAS AND A. N. VENETSANOPOULOS, *Nonlinear Digital Filters: Principles and Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.
- [28] D. QI, F. GUO, AND L. YU, *Medical image edge detection based on omnidirectional multi-scale structure element of mathematical morphology*, in Proceedings of the IEEE International Conference on Automation and Logistics, IEEE, Washington, DC, 2007, pp. 2281–2286.
- [29] L. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithm*, Phys. D, 60 (1992), pp. 259–268.
- [30] J. SERRA, *Image Analysis and Mathematical Morphology*, Academic Press, New York, 1982.
- [31] C. E. SHANNON, *Communication in the presence of noise*, Proc. Institute of Radio Engineers, 37 (1949), pp. 10–21.
- [32] J. STARCK, F. MURTAGH, AND J. FADILI, *Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity*, Cambridge University Press, Cambridge, UK, 2010.
- [33] W. STEFAN, R. A. RENAUT, AND A. GELB, *Improved total variation-type regularization using higher order edge detectors*, SIAM J. Imaging Sci., 3 (2010), pp. 232–251.
- [34] E. TADMOR AND J. ZOU, *Three novel edge detection methods for incomplete and noisy spectral data*, J. Fourier Anal. Appl., 14 (2008), pp. 744–763.
- [35] B. TIAN, H. YUAN, AND X. YUE, *Feature extraction algorithm for space targets based on fractal theory*, in Proceedings of the Second International Conference on Space Information Technology, SPIE-INT SOC Optical Engineering, SPIE, Bellingham, WA, 2007, p. 79518.
- [36] J. TRZASKO, A. MANDUCA, AND E. BORISCH, *Highly undersampled magnetic resonance image reconstruction via homotopic  $\ell_0$ -minimization*, IEEE Trans. Medical Imaging, 28 (2009), pp. 106–121.
- [37] N. VASWANI AND W. LU, *Modified-cs: Modifying compressive sensing for problems with partially known support*, IEEE Trans. Signal Process., 58 (2010), pp. 4595–4607.
- [38] Y. WANG, J. YANG, W. YIN, AND Y. ZHANG, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imaging Sci., 1 (2008), pp. 248–272.
- [39] Y. WANG AND W. YIN, *Sparse signal reconstruction via iterative support detection*, SIAM J. Imaging Sci., 3 (2010), pp. 462–491.

- [40] J. YANG, W. YIN, Y. ZHANG, AND Y. WANG, *A fast algorithm for edge-preserving variational multi-channel image restoration*, SIAM J. Imaging Sci., 2 (2008), pp. 569–592.
- [41] J. YANG, Y. ZHANG, AND W. YIN, *An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise*, SIAM J. Sci. Comput., 31 (2009), pp. 2842–2865.
- [42] J. YANG, Y. ZHANG, AND W. YIN, *A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data*, IEEE J. Selected Topics Signal Process., Special Issue on Compressed Sensing, 4 (2010), pp. 288–297.
- [43] W. YIN, S. OSHER, D. GOLDFARB, AND J. DARBON, *Bregman iterative algorithms for  $\ell_1$ -minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168.
- [44] L. ZHAI, S. DONG, AND H. MA, *Recent methods and applications on image edge detection*, in IEEE International Workshop on Geoscience and Remote Sensing, IEEE, Washington, DC, 2008, pp. 332–335.
- [45] L. ZHANG, A. BUTLER, AND C. SUN, *Fractal dimension assessment of brain white matter structural complexity post stroke in relation to upper-extremity motor function*, Brain Research, 1228 (2008), pp. 229–240.
- [46] M. V. W. ZIBETTI AND A. R. DE PIERRO, *Separate magnitude and phase regularization in MRI with incomplete data: Preliminary results*, in Proceedings of the IEEE International Symposium on Biomedical Imaging, IEEE, Washington, DC, 2010, pp. 736–739.
- [47] D. ZIOU AND S. TABBONE, *Edge detection techniques—an overview*, Internat. J. Pattern Recognition Image Anal., 8 (1998), pp. 537–559.