Mathematical Foundations of Computing
©American Institute of Mathematical Sciences
Volume **3**, Number **4**, November **2020**

doi:10.3934/mfc.2020025

pp. **229–247**

INPAINTING VIA SPARSE RECOVERY WITH DIRECTIONAL CONSTRAINTS

Xuemei Chen*

Department of Mathematics and Statistics
University of North Carolina Wilmington
Wilmington, NC 28403, USA

Julia Dobrosotskaya

Department of Mathematics, Applied Mathematics and Statistics
Case Western Reserve University
Cleveland, OH 44106, USA

Abstract. Image inpainting is a particular case of image completion problem.We describe a novel method allowing to amend the general scenario of using sparse or TV-based recovery for inpainting purposes by an efficient use of adaptive one-dimensional directional "sensing" into the unknown domain. We analyze the smoothness of the image near each pixel on the boundary of the unknown domain and formulate linear constraints designed to promote smooth transitions from the known domain in the directions where smooth behavior have been detected. We include a theoretical result relaxing the widely known sufficient condition of sparse recovery based on coherence, as well as observations on how adding the directional constraints can improve the well-posedness of sparse inpainting.

The numerical implementation of our method is based on ADMM. Examples of inpainting of natural images and binary images with edges crossing the unknown domain demonstrate significant improvement of recovery quality in the presence of adaptive directional constraints. We conclude that the introduced framework is general enough to offer a lot of flexibility and be successfully utilized in a multitude of image recovery scenarios.

1. **Introduction.** The task of recovering missing data elements from the known part of the dataset in a variety of contexts is typical yet often ill posed problem in many fields of scientific research. *Inpainting*, the process of reconstructing lost or occluded parts of images, falls into this category, and is, in fact, an ancient art itself.

The human visual system has an amazing ability to fill in the missing parts of images in order to complete our visual perception and better orient us in space, but automatizing this process is very far from trivial. In other scenarios, depending on the types of data, a deep neural network resembling the human senses may not be able to fill in the gaps, such as missing parts in X-Ray or magnetic resonance imaging (MRI) due to the properties of data by far exceeding the complexity of

2020 *Mathematics Subject Classification.* Primary: 65F45, 65K10; Secondary: 94A16.

Key words and phrases. Inpainting, sparsity, compressed sensing, adaptive, directional.

The first author is supported by NSF DMS-2050028.

* Corresponding author: Xuemei Chen.

even our dymanic visual perception. So automating the filling process, or digital inpainting, is necessary in these situations.

Three types of inpainting methods have been widely used and studied. The first type concerns diffusion-based approaches via PDE or variational methods. The use of PDE methods, which propagate linear structures or level lines into the missing region was pioneered by Bertalmio et al [4]. Shen and Chan [14] proposed a variational framework based on total variation (TV) to recover the missing domain. Many other PDE-based and variational methods showed great recovery results in a variety of scenarios([3, 6, 5] and more). The second type is examplar-based image inpainting, which sample and copy best matching texture patches from the known image neighborhood to the unknown region [20, 15, 32], which is motivated to overcome the smoothing effect in the textured region or larger missing region. The third type is a frame-based method [21, 7, 9, 23, 17] which represents images in sparsifying frames like wavelet frames.

The recent success of compressed sensing (CS) in sparse data recovery [19, 12] such as matrix completion [11] and image restoration [24, 31] has led to a growing interest in applying and developing CS methods for imaging data. Moreover, some nonlinear PDE related variational methods can be computationally complicated, whereas more straightforward algorithms are developed for CS as demonstrated in [9]. Therefore it adds simplicity and efficiency to reduce an inpainting problem to a basis-pursuit problem.

Often in order to build a successful model of sparse image reconstruction one needs to have either a very redundant or data adaptive representation system. Series of successful designs of data driven wavelet/tight frame representations have been developed in [22, 8, 2, 18, 28]. In [8], the authors developed a new approach to construct adaptively learned discrete tight frames that are more likely to give a highly sparse approximation of the local features, which is illustrated by applying those for image denoising. A few of those approaches assume joint sparsity of the patches of the image in the recovered frame, or otherwise exploit the notion of joint sparsity. In case of image inpainting such an assumption might present difficulties due to nature of the unknown domain.

Authors in [22] present an elegant data adaptive construction in the composite wavelet or shearlet context, where the data adaptive representations are produced via creating an adaptive multiresolution analysis. However, due to the necessity of translation invariance in imaging the transforms in question must be undecimated (stationary). This, together with the use of multiple shearing filters creates a very significant level of redundancy, and thus is computationally heavy.

Our method provides middle ground between sparse recovery via data adaptive vs generic systems. This new technique upgrades the sparse recovery in well established representation systems in which the natural images are typically sparse by augmenting the sensing matrix with a data adaptive part imposing directional constraints, more precisely - the requirements of selective smooth transitions into the unknown domain along the directions inferred from the known parts of the image. In this sense our method can be viewed as an exemplar-based method in a generalized sense, as the information in the known domain around the boundary is prioritized and we selectively enforce the priority information to be propagated into the unknown domain.

It is important to notice, that while we traditionally assume wavelets and associated families of functions to provide sparse representation for the natural images, or images with repetitive texture and piecewise smooth edges, these assumptions typically fail in cases of images with resolution below 256 by 256, thus the classical sparse reconstruction criteria need to be adjusted for a meaningful formulation of the respective inverse problem. Moreover, in problems with severe data loss where the data recovery should rely solely on the geometric features of the image information in the nearest proximity, our approach allows a straightforward way to amend any existing recovery technique with directional constraints that allow an adaptive choice of the propagation radius and smoothness of transitions defined by the parameter (filter) choices.

The framework we describe is very flexible and can be used with a wide variety of representations, it also allows to incorporate a TV minimization component in case of appropriate image properties and unknown domains. In fact, we propose a numerical implementation which, in case only TV and directional components are involved, performs inpainting within seconds.

We describe the general framework of the method in Section 2, within which Section 2.3 describes the formation of the directional constraints in detail. Section 3 is devoted to the details of the numerical implementation of the algorithm in the general framework and the intuition behind the choice of the parameters defining the directional constraints. Section 5 demonstrates particular examples of the numerical experiments, highlighting features of the method. We include a novel sufficient condition for the sparse recovery problem in Section 4. Though not immediately relevant to the numerical examples shown, we believe it is an important observation that emphasizes the significance of considering not only the immediate upper and lower bounds of coherence, but also the dynamics of the progression from upper to lower. Directions of future work are described in Section 6.

2. **Main idea of the method.**

2.1. **Notations.**

We will use the variables x or X to represent an image as a matrix: $X \in \mathbb{R}^{m\times n}$ or as a column vector $x \in \mathbb{R}^d = \mathbb{R}^{mn}$. We will use the following notation for their conversion when necessary: $\operatorname{vec}(X) = x, \operatorname{vec}^{-1}(x) = X$. Let I_d denote the identity matrix of size $d \times d$. For $x \in \mathbb{R}^d$, we define $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$ for $p \geq 1$. The notation $[N]$ is for the set $\{1, 2, \cdots, N\}$. Given an index set $T \subset [N]$, let T^c denote the complement index set, and x_T denote the restriction of x to the index subset T. A vector is s-sparse if at most s of its coordinates are nonzero.

Let ∇_1, ∇_2 be the horizontal and vertical forward difference operator respectively. They are defined as

$$(\nabla_1(X))_{i,j} = X_{i,j} - X_{i,j-1}, \qquad (\nabla_2(X))_{i,j} = X_{i,j} - X_{i-1,j},$$

where the edges are handled periodically. We define the discrete gradient of an $m\times n$ image X using the discrete derivative as $\nabla X = (\nabla_1 X, \nabla_2 X)^T$, and the discretized TV seminorm as

$$\|X\|_{TV} := \|\nabla X\|_1 = \|\nabla_1 X\|_1 + \|\nabla_2 X\|_1 = \sum_{i,j} |(\nabla_1 X)_{i,j}| + \sum_{i,j} |(\nabla_2 X)_{i,j}|.$$

Note that our definition gives the anisotropic version of TV [16].

In the finite dimensional setting, a frame [13, 30] can be defined as a collection of vectors that span $\mathbb{R}^d$. We will further conventionally use F to denote a frame, i.e. a matrix whose columns form a frame. The frame F is a *Parseval* (tight) frame if and only if $FF^* = I_d$.

Given $\lambda > 0$, the *soft thresholding* operator is defined as $S_\lambda(a) = \max\{0, a - \lambda\} - \max\{0, -a-\lambda\}$ for any $a \in \mathbb{R}$. $S_\lambda(\cdot)$ can also applied to a vector as it applies component-wise. It is well known that

$$S_\lambda(v) = \arg\min_x \frac{1}{2}\|x - v\|_2^2 + \lambda\|x\|_1. \tag{1}$$

2.2. **Model overview.** Consider a grayscale image x_0 with some missing area, see Figure 1. Let Λ be the set of indices corresponding to the pixels in the known area, and Λ^c - the indices of the missing area. We wish to recover the missing part of the image based on the known part.

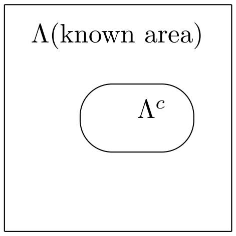


Figure 1. An image to be inpainted

Let P_Λ be the projection onto the known domain Λ, that is, P_Λ is the matrix obtained by zeroing out the rows of I_d corresponding to Λ^c.

The TV regularization technique is a simple and effective image recovery tool [14] that originates from the ROF denoising model [29]. It can be used for inpainting in the following formulation

$$\min_x \|\nabla_1 x\|_1 + \|\nabla_2 x\|_1, \quad \text{subject to } P_\Lambda x = P_\Lambda x_0, \tag{2}$$

or, in the unconstrained form,

$$\min_x \lambda\left(\|\nabla_1 x\|_1 + \|\nabla_2 x\|_1\right) + \frac{1}{2}\|P_\Lambda x - P_\Lambda x_0\|_2^2. \tag{3}$$

Given a frame F, the frame-based sparse recovery approach (analysis version) can be modeled as

$$\min \|F^* x\|_1, \quad \text{subject to } P_\Lambda x = P_\Lambda x_0, \tag{4}$$

or, in the unconstrained form,

$$\min \mu\|F^* x\|_1 + \frac{1}{2}\|P_\Lambda x - P_\Lambda x_0\|_2^2, \tag{5}$$

under the assumption that $F^* x$ is approximately sparse [7, 9].

Our method combines (3) and/or (5) with an additional linear system $\tilde{B}x = 0$. In other words, we will solve

$$\min_x \lambda\left(\|\nabla_1 x\|_1 + \|\nabla_2 x\|_1\right) + \mu\|F^* x\|_1 + \frac{1}{2}\|P_\Lambda x - P_\Lambda x_0\|_2^2 + \frac{\beta}{2}\|\tilde{B}x\|_2^2. \tag{6}$$

The additional constraint $\tilde{B}x = 0$ implements our intention to preserve smoothness along certain line segments crossing the boundary of the unknown domain. It is implemented by taking the dot products of the corresponding vectors of intensity values at the chosen pixels with a certain filter. For this reason, $\tilde{B}x = 0$ only involves a small subset of Λ^c, which we call active unknowns, denoted by U. After solving (6), we only use values of the minimizer at pixels/indices $U \subset \Lambda^c$. Consequently, Λ^c is shrunk and we keep solving (6) until $\Lambda^c = \emptyset$. Next section provides the details of the process to create $\tilde{B}x = 0$.

2.3. **Adaptive directional constraints.** We define a boundary pixel/index to be a pixel that is in the unknown domain Λ^c and has an adjacent pixel that is in Λ. Each pixel has 8 adjacent pixels. The linear system $\tilde{B}x = 0$ is generated by testing the smoothness of the image near the boundary of the unknown domain and imposing the constraints that the smooth transitions in certain directions must be preserved across the boundary of the domain as well. For each boundary pixel, there are 8 choices of directions $\{k\pi/4\}_{k=0}^{7}$ for consideration. Depending on the geometry of the domain and features of the known part of the image only few of them will end up being considered.

Let $h = (h_1, h_2, \cdots, h_l)$ denote a high pass filter. For simplicity of the illustration, we will assume image intensity to have the range $0, \ldots, 255$ in this discussion. Numerical experiments were performed with normalized images.

1. **Testing the known part of the image.** For each pixel $X_{i,j}$ of the unknown domain we check whether there are l pixels in the known domain along a straight line segment starting from a pixel adjacent to $X_{i,j}$ pixel in each of the 8 possible directions. For instance, for the direction associated with $\pi/4$ or 'North East' we would need to check if pixels $\{X_{i+t,j+t}\}_{t=1}^{l}$ are in the known domain Λ. In general, the respective vectors are formed by considering $\{X_{i+\delta_1 t, j+\delta_2 t}\}_{t=1}^{l}$ with $\delta_1, \delta_2 \in \{-1, 0, 1\}$, namely - direction 0, or 'East' corresponds to $\delta_1 = 1$, $\delta_2 = 0$, direction $\pi/4$, or 'North East' corresponds to $\delta_1 = 1$, $\delta_2 = 1$, and so on up to direction $7\pi/4$, or 'South East' with $\delta_1 = 1$, $\delta_2 = -1$. For those directions where we found enough pixels (i.e. indices $\{(i + \delta_1 t, j + \delta_2 t)\}_{t=1}^{l}$ correspond to pixels in Λ), we perform analysis of smoothness of the image in that direction near the unknown pixel $X_{i,j}$ by computing the dot product between those intensity values and the wavelet high pass filter h. If the product $\sum_{t=1}^{l} h_k X_{i+\delta_1 t, j+\delta_2 t} = 0$ or is sufficiently close to zero, we assume that this direction can be used to form a directional sensing equation.

2. **Forming the constraints.** For those directions that were chosen we form a vector from the $l/2 + \alpha$ known pixels and $l/2 - \alpha$ unknown pixels along a straight line segment through the boundary pixel $X_{i,j}$ and state the directional constraint: the dot product of this vector with the filter h must be 0. In our experiments we typically use $\alpha = 0$ or $\alpha = 1$. Taking, for simplicity, $\alpha = 0$, we get the equations of the form $\sum_{k=1}^{l} h_k X_{i+\delta_1(k-l/2), j+\delta_2(k-l/2)} = 0$. We will call the equations formed with of $\alpha = 0$ *centered* and the ones with $\alpha > 0$ *shifted*. Among all equations, we can pick exactly one equation per boundary pixel, corresponding to the direction with the smallest dot product at the testing stage, randomly picking one in cases of multiple minima. If one wishes to keep the constraints balanced in terms of having splitting the energy evenly between the known and unknown domain, it would make sense to require $\alpha = 0$ (exactly $l/2$ known and $l/2$ unknown pixels present in each constraint). More discussions on forming the constraint equations can be found in Section 2.3.1.

We will use a 10×10 image as an example to illustrate this idea in detail. In Figure 2, a binary image (pixel value 255 is marked, 0 is left as blank) is missing a 4×4 square, where the unknown coordinates are labeled. In this example, we pick $h = (\sqrt{3} - 1, 3 - \sqrt{3}, -3 - \sqrt{3}, 1 + \sqrt{3})/8 = (h_1, h_2, h_3, h_4)$, which is the Daubechies high frequency wavelet of length 4. We will pick one equation per boundary pixel and use $\alpha = 0$.

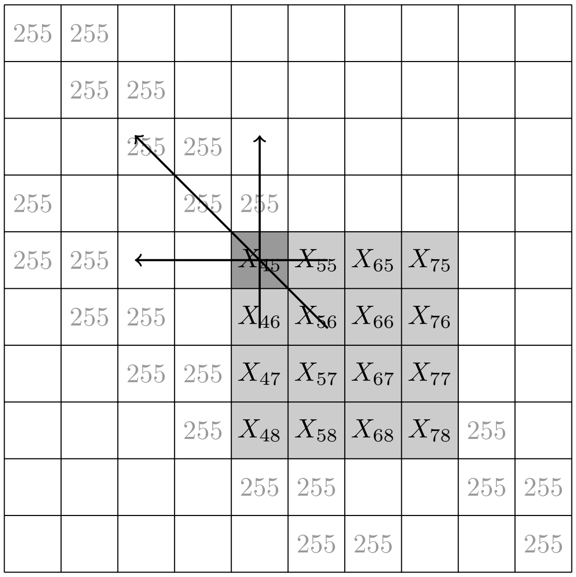

255 255
255 255
255 255
255 255 255
255 255 X_{45} X_{55} X_{65} X_{75}
255 255 X_{46} X_{56} X_{66} X_{76}
255 255 X_{47} X_{57} X_{67} X_{77}
255 X_{48} X_{58} X_{68} X_{78} 255
255 255 255 255
255 255 255

FIGURE 2. Forming boundary equations

Take the boundary pixel X_{45} as an example, direction 0 is omitted since all pixels along that direction are in Λ^c ; direction $\pi/4$ is omitted since it has 3 pixels in Λ and 1 in Λ^c; directions $5\pi/4, 6\pi/4, 7\pi/4$ are omitted for the same reasons; only directions $\pi/2, 3\pi/4, \pi$ are viable (as marked) and the potential constraints are

$$\begin{aligned}
&\text{Direction } \pi/2: && X_{46}h_1 + X_{45}h_2 + 255h_3 + 0h_4 = 0\\
&\text{Direction } 3\pi/4: && X_{56}h_1 + X_{45}h_2 + 255h_3 + 255h_4 = 0\\
&\text{Direction } \pi: && X_{55}h_1 + X_{45}h_2 + 0h_3 + 0h_4 = 0
\end{aligned}$$

Observing the image in Figure 2, we ideally would like to propagate the pixel values along the $3\pi/4$ direction to X_{45}, which means that it would be the best if we pick the $3\pi/4$ direction. In order to achieve this, we extend along each direction by $l/2$ pixels (See Figure 3). Now we have l known pixels along each of the three directions for X_{45}, and we can compute their inner products with the same filter h:

$$\begin{aligned}
&\text{Direction } \pi/2: && 255h_1 + 0h_2 + 0h_3 + 0h_4 = 23.33\\
&\text{Direction } 3\pi/4: && 255h_1 + 255h_2 + 255h_3 + 255h_4 = 0\\
&\text{Direction } \pi: && 0h_1 + 0h_2 + 255h_3 + 255h_4 = -63.75
\end{aligned}$$

The direction with the smallest inner product (in magnitude) is picked, so in this example, direction $3\pi/4$ is the winner, as desired. For the boundary pixel X_{45}, we obtain the linear equation

$$h_1X_{56} + h_2X_{45} = -255h_3 - 255h_4, \tag{7}$$

or equivalently

$$h_1X_{56} + h_2X_{45} + h_3X_{34} + h_4X_{23} = 0. \tag{8}$$

This operation will be repeated for all 12 boundary pixels, and, if we decide to include only one equation per boundary pixel, $\tilde{B}$ is 12×100.

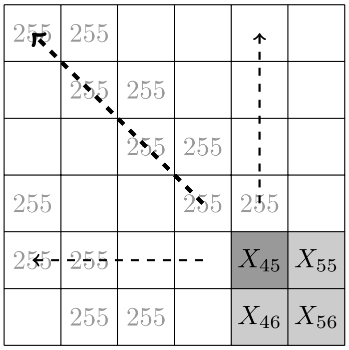


FIGURE 3. Forming boundary equations: pick the best direction

Equation (8) is one row in $\tilde{B}x = 0$. We see that (7) and (8) are equivalent, so in the case we use (7), it will be denoted as $Bx = b$.

2.3.1. *Choice of the filter and the number of equations per boundary pixel.* **Choice of the high pass Daubechies filter based on wavelet properties** In our experiments we use high pass wavelet filters from the Daubechies family. Our motivation is based on the properties of the DB-P wavelets to have smallest compact support for an associated level of regularity P, namely - the basis functions have P continuous derivatives and the wavelet function ψ has P vanishing moments, making it orthogonal to polynomials of degrees up to P ([26, 25]). The associated high pass filter is a vector of length $2P$, orthogonal to vectors of polynomial values (of degree up to P) with high order of accuracy. Thus, requiring the dot product of a vector intensities along a chosen direction with the high pass Daubechies filter of order P (length $2P$) to be equal to zero we effectively impose the condition of image values changing in this direction with at most polynomial rate of controlled degree.

Another explanation can be given using the fact that a high pass Daubechies P filter contains coefficients of a finite difference approximations of the derivative of the respective order P, we are requiring that the change of image intensities in this direction have smoothness of order P. Thus, using the Haar filter simply allows us to impose constant transition to the unknown domain in a chosen direction.

Choice of the filter based on the properties of the unknown domain. Since our choice of directions for the additional directional constraints is preceded by analyzing the known part of the image in the respective direction near the boundary, one needs to make sure that the known domain contains enough pixels in that direction. Thus, for non-convex domains with complicated boundary it is appropriate to choose shorter filters, such as Daubechies 2. However, in case of a large convex domain, using longer filters seems more appropriate.

Let us notice that if we only include directions for which the analysis of the known domain gave a result sufficiently close to zero, it makes sense to include multiple filters of varying length, thus automatically making the requirement of transition smoothness adaptive to a particular image.

An alternative choice is to use the complex filter of length l defined as $h = \sqrt{l}[1, e^{2\pi i/l} \dots e^{2\pi i(l-1)/l}]$. The length of the vector can be arbitrary and it will impose constant (i.e. zero frequency) transitions across the domain in the chosen directions.

Choice of the number of equations per boundary pixel. Depending on the image configuration, we can pick exactly one equation per boundary pixel, corresponding to the direction with the smallest dot product at the testing stage. Alternatively, we can pick p directions with smallest products, or only include an equation corresponding to a certain direction if the testing dot product was very close to zero.

2.3.2. *Illustration of parameter choices in a simplified scenario.* In case of thin domains, using long enough filters allows to perform a heuristic inpainting solely based on the directional sensing. This will allow us to clearly see which choices of the constraints are more appropriate for a chosen domain.

Figure 4 shows for a simple 64×64 image (top half white, bottom half black) with a missing domain of size 4×29, it is possible to create sufficiently many directional equations involving all unknown pixels so that the problem can be solved as an overcomplete problem in the least squares sense.

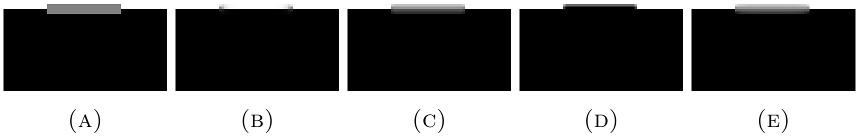


FIGURE 4. Recovery via directional sensing only. (a) Image with a thin missing domain. (b) 'DB-2', all smooth directions per pixel included, both centered and shifted equations used. (c) 'DB-2', all smooth directions per pixel included, only centered equations used. (d) 'DB-2', one direction per pixel included, only centered equations used could be formed as the filter has length 2. (e) 'DB-4', all smooth directions included, both centered and shifted equations used.

Figure 4 (b) shows the result of imposing the constraints using the high pass filter 'DB-2' (of length 4), all smooth directions per pixel included, equations formed with both centered and shifted filters (i.e. 2 known-2 unknown and 3 known 1 unknown) combinations considered (see part two of the algorithm description in Section 2.3). We observe that the corner points of the missing domain for which three directions were used ended up with undesirable recovered intensity due to contradictory conditions. In Figure 4 (c) we can see the changes in the result if only centered equations are used, (d) gives us an underdetermined system since we only choose one equation per pixel, a similar picture would result if we used a short filter DB-1, and thus the equations would not involve all unknown pixels. (e) uses 'DB-4', which is too long for the chosen domain, creates overdetermined system of conflicting constrains resulting in a blurred averaged recovery.

The above tests simply give a visual meaning to the impact of the directional constraints as our choice of the filter and number and type of the equations vary. However, the directional constraints only serve as a useful addition to the sparse image recovery model. Figure 5 shows examples of the directional constraints that were used in our actual numerical experiments involving sparse recovery that are presented in Section 5.

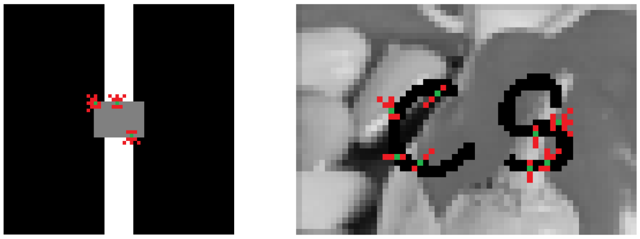


FIGURE 5. Examples of directional constraints adaptively formulated for the inpainting examples discussed further in the text. In both cases the 'DB2' filter of length 4 was used to form centered equations. Green dots indicate the unknown pixels around which the constraints were formed. Red dots indicate other pixels present in those constraints. The constraints are shown only for some pixels to avoid overcrowding the images. The actual results of inpainting appear later in Figures 6 and 9.

3. **Numerical implementation of minimization using ADMM.** The problem (6) is solved via the alternating direction method of multiplier (ADMM) [27]. In this section, we will use $Bx = b$ instead of $\tilde{B}x = 0$ (see the paragraph after (8)) to maximize sparsity of the coefficient matrix. The columns of B correspond to Λ are all zeros columns. Indeed, if we let $x = \begin{bmatrix} x_\Lambda \\ x_{\Lambda^c} \end{bmatrix}$, then $B = [B_\Lambda, B_{\Lambda^c}] = [0, B_{\Lambda^c}]$. Therefore $Bx = B_{\Lambda^c} x_{\Lambda^c}$.

We first convert (6) to

$$\min \lambda\|d_1\|_1 + \lambda\|d_2\|_1 + \mu\|f\|_1 + \frac{1}{2}\|P_\Lambda x - P_\Lambda x_0\|_2^2 + \frac{\beta}{2}\|Bx - b\|_2^2$$

$$\text{s.t. } d_1 = \nabla_1 x, d_2 = \nabla_2 x, f = F^* x.$$

Its Augmented Lagrange is

$$L(x, d_1, d_2, f, r_1, r_2, r_f) = \frac{1}{2}\|P_\Lambda x - P_\Lambda x_0\|_2^2 + \frac{\beta}{2}\|Bx - b\|_2^2 + \frac{\rho_f}{2}\|F^* x - f$$
$$+ r_f\|_2^2 + \frac{\rho}{2}\|\nabla_1 x - d_1 + r_1\|_2^2 + \frac{\rho}{2}\|\nabla_2 x - d_2 + r_2\|_2^2 + \lambda\|d_1\|_1 + \lambda\|d_2\|_1 + \mu\|f\|_1.$$

With a pre-selected ρ_f, ρ and initializations $x^0, d_1^0, d_2^0, f^0, r_1^0, r_2^0, r_f^0$, the iterations for solving (6) are

$$x^{k+1} = \arg\min_x L(x, d_1^k, d_2^k, f^k, r_1^k, r_2^k, r_f^k) \tag{9}$$

$$\begin{aligned} d_1^{k+1} &= \arg\min_x L(x^{k+1}, d_1, d_2^k, f^k, r_1^k, r_2^k, r_f^k) \\ &= \arg\min_{d_1} \lambda\|d_1\|_1 + \frac{\rho}{2}\|\nabla_1 x^{k+1} - d_1 + r_1^k\|_2^2 \end{aligned} \tag{10}$$

$$\begin{aligned} d_2^{k+1} &= \arg\min_x L(x^{k+1}, d_1^k, d_2, f^k, r_1^k, r_2^k, r_f^k \\ &= \arg\min_{d_2} \lambda\|d_2\|_1 + \frac{\rho}{2}\|\nabla_2 x^{k+1} - d_2 + r_2^k\|_2^2 \end{aligned} \tag{11}$$

$$\begin{aligned} f^{k+1} &= \arg\min_f L(x^{k+1}, d_1^k, d_2^k, f, r_1^k, r_2^k, r_f^k) \qquad (12)\\ &= \arg\min_f \mu\|f\|_1 + \frac{\rho_f}{2}\|F^*x^{k+1} - f + r_f^k\|_2^2 \end{aligned}$$

$$r_1^{k+1} = r_1^k + \nabla_1 x^{k+1} - d_1^{k+1} \qquad (13)$$

$$r_2^{k+1} = r_2^k + \nabla_2 x^{k+1} - d_2^{k+1} \qquad (14)$$

$$r_f^{k+1} = r_f^k + F^* x^{k+1} - f^{k+1} \qquad (15)$$

These seven steps are standard in ADMM. See [27, Section 3.1].

The problem (9) is to find the least squares solution of

$$\begin{bmatrix} P_\Lambda \\ \sqrt{\beta}B \\ \sqrt{\rho_f}F^* \\ \sqrt{\rho}\nabla_1 \\ \sqrt{\rho}\nabla_2 \end{bmatrix} x = \begin{bmatrix} P_\Lambda x_0 \\ \sqrt{\beta}b \\ \sqrt{\rho_f}f^k - \sqrt{\rho_f}r_f^k \\ \sqrt{\rho}d_1^k - \sqrt{\rho}r_1^k \\ \sqrt{\rho}d_2^k - \sqrt{\rho}r_2^k \end{bmatrix},$$

whose normal equation is

$$\begin{aligned} &(P_\Lambda^* P_\Lambda + \beta B^*B + \rho\nabla_1^*\nabla_1 + \rho\nabla_2^*\nabla_2 + \rho_f FF^*)x \\ =&P_\Lambda^* P_\Lambda x_0 + \beta B^* b + \rho\nabla_1^*(d_1^k - r_1^k) + \rho\nabla_2^*(d_2^k - r_2^k) + \rho_f F(f^k - r_f^k) \qquad (16) \end{aligned}$$

The problems (10), (11), and (12) have a direct solution. By (1), each one is solved by a soft thresholding step:

$$\begin{cases} d_1^{k+1} = & S_{\frac{\lambda}{\rho}}(\nabla_1 x^{k+1} + r_1^k) \\ d_2^{k+1} = & S_{\frac{\lambda}{\rho}}(\nabla_2 x^{k+1} + r_2^k) \\ f^{k+1} = & S_{\frac{\mu}{\rho}}(F^* x^{k+1} + r_f^k) \end{cases} \qquad (17)$$

3.1. **Solving the normal equation.** Equation (16) is the most time consuming step. If an image is $m \times n$, then the coefficient matrix is $mn \times mn$, which is very big for a relatively high resolution image. Fortunately, we can take advantage of the special structures of these operators, and solve (16) efficiently. We explain the details below.

First, P_Λ is a diagonal matrix. We use E_Λ to indicate the $m \times n$ mask where its value is 1 on Λ and 0 on Λ^c. Then $P_\Lambda^* P_\Lambda x = P_\Lambda x = E_\Lambda \odot X$, where $\odot$ is the point-wise multiplication of two matrices.

Secondly, $B = [0, B_{\Lambda^c}]$, so $B^*B = \begin{bmatrix} 0 & 0 \\ 0 & B_{\Lambda^c}^* B_{\Lambda^c} \end{bmatrix}$, and $B^*Bx = \begin{bmatrix} 0 \\ B_{\Lambda^c}^* B_{\Lambda^c} x_{\Lambda^c} \end{bmatrix}$, which is computed on the scale of number of unknown pixels, not the whole image. We also note that B^*B is diagonally dominant. Let $\text{diag}(B^*B)$ be the $mn \times mn$ diagonal matrix that only extracts the diagonal of B^*B, and B_D be the diagonal of B^*B that reshapes to an $m \times n$ matrix, then $\text{diag}(B^*B)x = \begin{bmatrix} 0 \\ \text{diag}(B_{\Lambda^c}^* B_{\Lambda^c}) x_{\Lambda^c} \end{bmatrix} = B_D \odot X$.

Thirdly, it can be easily shown that

$$\nabla_1(x) = XD_n, \quad \nabla_1^*(x) = XD_n^*, \quad \nabla_2(x) = D_m^* X, \quad \nabla_2^*(x) = D_m X,$$

where

$$D_k = \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & 1 & -1 & & \\ & & & \ddots & & \\ & & & & 1 & -1 \\ -1 & & & & & 1 \end{bmatrix}_{k\times k}.$$

Moreover, $\nabla_1^*\nabla_1 + \nabla_2^*\nabla_2$ is diagonally dominant whose diagonal entries are all 4.

Lastly, we will assume F is Parseval so FF^* is the identity.

Given a system $\Phi x = a$, one iteration of Jacobi's method is

$$\mathrm{diag}(\Phi)x^{k+1} = a - (\Phi - \mathrm{diag}(\Phi)x^k),$$

which is to separate Φ to the diagonal part and non-diagonal part.

Since the system (16) is diagonally dominant, we will use Jacobi's iteration to solve it approximately. Let the right hand side of (16) be rhs, then (16) becomes

$$\left(\underbrace{P_\Lambda + \beta\mathrm{diag}(B^*B) + 4\rho + \rho_f}_{diagonal\ part} + \underbrace{\beta[B^*B - \mathrm{diag}(B^*B)] + \rho[\nabla_1^*\nabla_1 + \nabla_2^*\nabla_2 - 4]}_{nondiagonal\ part}\right) x = \mathrm{rhs}.$$

According to Jacobi's iteration, this can be approximately solved by

$$(P_\Lambda + \beta\mathrm{diag}(B^*B) + 4\rho + \rho_f)x^{k+1} = \mathrm{rhs} - \beta[B^*B - \mathrm{diag}(B^*B)]x^k - \rho[\nabla_1^*\nabla_1 + \nabla_2^*\nabla_2 - 4]x^k, \tag{18}$$

which is simplified to

$$\begin{aligned}&[E_\Lambda + \beta B_D + 4\rho E_{m\times n} + \rho_f E_{m\times n}] \odot X^{k+1} \\ &= \mathrm{rhs} - \beta\mathrm{vec}^{-1}(B^*Bx^k) + \beta B_D \odot X^k - \rho(X^kD_nD_n^* + D_mD_m^*X^k) + 4\rho X^k, \end{aligned}\tag{19}$$

where $E_{m\times n}$ is the $m \times n$ matrix whose entries are all 1's. (19) can be solved very efficiently.

With a fixed ρ, ρ_f, the iterative steps for solving (6) can be summarized below.

```
Initialization:   $x^0, d_1^0, d_2^0, w^0, r_1^0, r_2^0, r_w^0, \rho, \rho_f$, tol
while             $\|x^k - x^{k-1}\|_2/\|x_0\|_2 >$ tol
                  perform (19) to obtain $x^{k+1}$,
                  $d_1^{k+1} = S_{\frac{\lambda}{\rho}}(\nabla_1 x^{k+1} + r_1^k)$,
                  $d_2^{k+1} = S_{\frac{\lambda}{\rho}}(\nabla_2 x^{k+1} + r_2^k)$,
                  $f^{k+1} = S_{\frac{\mu}{\rho}}(F^* x^{k+1} + r_f^k)$,
                  $r_1^{k+1} = r_1^k + \nabla_1 x^{k+1} - d_1^{k+1}$,
                  $r_2^{k+1} = r_2^k + \nabla_2 x^{k+1} - d_2^{k+1}$,
                  $r_f^{k+1} = r_f^k + F^* x^{k+1} - f^{k+1}$
end
```

We remark that in case F is an orthonormal basis and no TV is involved, we can solve it via soft shrinkage, but in its most complete formulation we need to use ADMM.

4. Sparse recovery: ℓ_1 minimization analysis.

Given a linear system $Ax = y = Ax_0$ and a frame F, solving the ℓ_1-analysis problem

$$\min \|F^*x\|_1, \quad \text{subject to } Ax = Ax_0 \tag{20}$$

tends to provide a solution that is sparse under F given A is well conditioned [10, 1]. In the compressed sensing framework, to guarantee a unique sparse recovery via such ℓ_1-analysis, there are sufficient conditions based on requirements on the null space of A (see Lemma 4.2), or a restricted isometry property of AF [10] conditional on the required level of sparsity. In this section, we will present another result of this kind. We believe this result is beneficial for the image inpainting context, and also a new theoretical result, to the author's best knowledge.

Definition 4.1. Let the columns of Φ be $\varphi_i, i \in [N]$. The coherence of Φ is defined as

$$\mu(\Phi) := \max_{i \neq j} \frac{|\langle \varphi_i, \varphi_j \rangle|}{\|\varphi_i\|_2^2}. \tag{21}$$

Given a positive integer $s \leq N-1$, we also define a general coherence function

$$\mu_s(\Phi) := \max_{j \in [N]} \max_{|T|=s, j \notin T} \sum_{i \in T} \frac{|\langle \varphi_i, \varphi_j \rangle|}{\|\varphi_i\|_2^2}.$$

It is clear that $\mu_1(A) = \mu(A)$.

The common definition of coherence requires the columns of Φ to be normalized, in which case it coincides with (21). However, normalizing columns hides the true coherence. We present the general version to accommodate all matrices and to reveal the true quantity that is important in signal recovery.

The following lemma can be viewed as a consequence of [1, Theorem 3.1]. Nevertheless we provide a proof as the theorem in [1] is too general.

Lemma 4.2. *Suppose for every $x \in \ker A \backslash \{0\}$, it holds that*

$$\|(F^*x)_T\|_1 < \|(F^*x)_{T^c}\|_1, \text{ for all } |T| \leq s. \tag{22}$$

Then for every vector x_0 such that F^*x_0 is s-sparse, (20) *has a unique solution x_0.*

Proof. Given x_0 such that F^*x_0 is s-sparse, let $Ax = Ax_0$ but $x \neq x_0$, then $v = x_0 - x \in \ker A \backslash \{0\}$. Suppose T is the support of F^*x_0, then

$$\begin{aligned}\|F^*x_0\|_1 &\leq \|F^*x_0 - (F^*x)_T\|_1 + \|(F^*x)_T\|_1 = \|(F^*v)_T\|_1 + \|(F^*x)_T\|_1 \\ &\overset{(22)}{<} \|(F^*v)_{T^c}\|_1 + \|(F^*x)_T\|_1 = \|(F^*x)_{T^c}\|_1 + \|(F^*x)_T\|_1 = \|(F^*x)\|_1,\end{aligned}$$

proving that in the feasible set, x_0 minimizes $\|F^*x\|_1$. □

Theorem 4.3. *Let F be a Parseval frame, and a positive integer $s \leq N$ be given. If*

$$\mu_s(AF) + \mu_{s-1}(AF) < 1, \tag{23}$$

then for every vector x_0 such that F^*x_0 is s-sparse, (20) *has a unique solution x_0.*

Proof. Using Lemma 4.2, we only need to show that (23) implies (22). Let $x \in \ker A \backslash \{0\}$, then $AFF^*x = Ax = 0$. Let $F^*x = (c_1, c_2, \cdots, c_N)$ and $AF = [v_1, v_2, \cdots, v_n]$, so $AFF^*x = 0$ translates into $\sum_j c_j v_j = 0$. Fix an arbitrary $|T| = s$, for every $i \in T$,

$$c_i \|v_i\|_2^2 = -\sum_{j \neq i} c_j \langle v_j, v_i \rangle = -\sum_{l \in T^c} c_l \langle v_l, v_i \rangle - \sum_{j \in T, j \neq i} c_j \langle v_j, v_i \rangle.$$

Then

$$|c_i| \|v_i\|_2^2 \leq \sum_{l \in T^c} |c_l| |\langle v_l, v_i \rangle| + \sum_{j \in T, j \neq i} |c_j| |\langle v_j, v_i \rangle|$$

$$\implies |c_i| \le \sum_{l \in T^c} |c_l| \frac{|\langle v_l, v_i\rangle|}{\|v_i\|_2^2} + \sum_{j \in T, j \neq i} |c_j| \frac{|\langle v_j, v_i\rangle|}{\|v_i\|_2^2}.$$

Summing over $i \in T$ and exchanging summations, we have

$$\begin{aligned}\|(F^*x)_T\|_1 &\le \sum_{l \in T^c} |c_l| \sum_{i \in T} \frac{|\langle v_l, v_i\rangle|}{\|v_i\|_2^2} + \sum_{j \in T} |c_j| \sum_{i \in T, i \neq j} \frac{|\langle v_j, v_i\rangle|}{\|v_i\|_2^2} \\ &\le \sum_{l \in T^c} |c_l| \mu_s(AF) + \sum_{j \in T} |c_j| \mu_{s-1}(AF) \\ &= \mu_s(AF)\|(F^*x)_{T^c}\|_1 + \mu_{s-1}(AF)\|(F^*x)_T\|_1,\end{aligned}$$

which simplifies to $(1 - \mu_{s-1}(AF))\,\|(F^*x)_T\|_1 \le \mu_s(AF)\|(F^*x)_{T^c}\|_1$. So (22) is satisfied if $1 - \mu_{s-1}(AF) > \mu_s(AF)$. □

A sufficient condition for (23) is $\mu(AF) < \frac{1}{2s-1}$ since $\mu_s(AF) \le s\mu(AF)$. We immediately have the following corollary.

Corollary 1. *Let F be a Parseval frame, and a positive integer $s \le N$ be given. If $\mu(AF) < \frac{1}{2s-1}$, then for every vector x_0 such that F^*x_0 is s-sparse, (20) has a unique solution x_0.*

The condition $\mu(AF) < \frac{1}{2s-1}$ is much easier to verify, but it may be too harsh for matrices to satisfy given a reasonably big s.

4.1. **Benefits of adding directional constraints.** This section analyzes the benefit of the directional constraint matrix $\tilde{B}$ in the context of our sparse recovery problem. In this analysis we set aside the TV term and only consider sparsity in a frame F. We will compare (4)

$$\min \|F^*x\|_1, \quad \text{subject to } P_\Lambda x = P_\Lambda x_0,$$

and

$$\min \|F^*x\|_1, \quad \text{subject to } \begin{bmatrix} P_\Lambda \\ \sqrt{\beta}\tilde{B} \end{bmatrix} x = \begin{bmatrix} P_\Lambda x_0 \\ 0 \end{bmatrix}, \tag{24}$$

or rather problem (6) with $\lambda = 0$ and non-zero β, and numerically test the sensing matrix improvement from P_Λ to $S = \begin{bmatrix} P_\Lambda \\ \sqrt{\beta}\tilde{B} \end{bmatrix}$.

It is clear that if the frame F has one or more elements supported entirely in the unknown domain, the problem (4) does not have properties like (22). This is the case for many scenarios involving large convex missing domains and typical sparse representations, say, compactly supported multiscale systems, which makes sparse recovery within those fixed systems ill posed.

We would like to point out that image inpainting is often subjective, and unique recovery via our model should not be a necessary requirement. Nevertheless, we illustrate numerically that the properties of the matrix SF indeed significantly improve (compared to $P_\Lambda F$) using our simple example of a binary image with a rectangular missing domain, even though the criteria of the lemmas mentioned above do not necessarily apply.

Table 1 lists the properties of $P_\Lambda F$ vs SF: maximum cosine of the angles between the columns, coherence μ_1, minimum and maximum singular values, which, when positive, make SF a frame. We fix F to be the 2D DB-4 wavelet transform matrix with 5 levels of decomposition, $\sqrt{\beta} = .75$. We use all smooth directions found, include only centered equations. The middle line of the table shows the improvement of including the directional restrictions obtained with DB-2 high pass filter h. The last line of the table represents the outcome of using two high pass filters, DB-2 and DB-4, and thus two matrices of directional constraints:

$$S_2 = \begin{bmatrix} P_\Lambda \\ \sqrt{\beta}\tilde{B} \\ \sqrt{\beta}\tilde{B}_2 \end{bmatrix}.$$

	Cosine	Coherence μ_1	Min singular value	Max singular value
$P_\Lambda F$	0.8541	3.8339	0	1
SF (1 filter)	0.4799	0.5888	0.0455	1.5255
S_2F (2 filters)	0.3913	0.4480	0.0937	1.9837

Table 1. Here $\sqrt{\beta} = .75$, F is the reshaped 2D DB-4 wavelet basis matrix, the first filter h is a high pass DB-2 filter, all smooth directions used and only centered equations included, the second filter added is a high pass DB-4 filter.

It is worth mentioning that using the orthonormal basis of Weyl matrices [30, Chapter 14] as F gives very low coherence of $P_\Lambda F$ even without the directional constraints, but adding those still allows to improve it - see Table 2. However, even though existence and uniqueness of sparse solutions are guaranteed, the results of recovery are not always great. While we can recover the image perfectly using the Weyl basis in case of the simple example we chose to test different directional filters, results of recovery of natural images, such as the peppers image in Figure 9a, are far from desired.

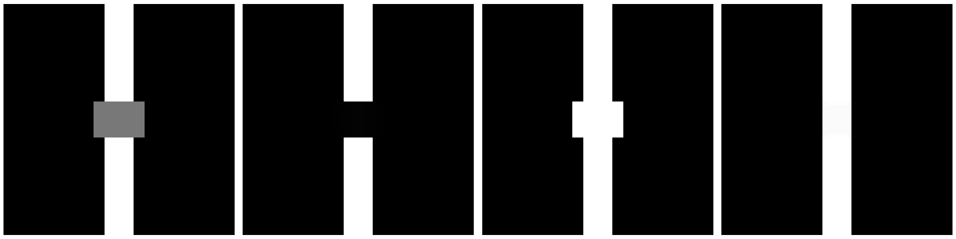
	Cosine	Coherence μ_1	Min singular value	Max singular value
$P_\Lambda F$	0.0667	0.0667	0	1
SF (1 filter)	0.0452	0.0457	0.0455	1.5255

Table 2. Here $\sqrt{\beta} = .75$, F is the reshaped 2D binary Weyl basis matrix, the filter h is a high pass DB-2 filter, all smooth directions used and only centered equations included.

5. **Numerical results.**

This section presents several numerical experiments on inpainting. The first three experiments only use TV regularization, that is, $\mu = 0$ in model (6), which is a natural choice for binary images with piecewise smooth edges. We compare recovery results with and without the directional information. Experiments are done on a Macbook Pro with 3.1GHz Intel Core i7 and 16GB memory.

The first experiment (Figure 6) is to remove a rectangular block from a 64×64 binary image with a vertical stripe in the middle. We pick a wide range of λ, but without the directional information (see Figure 6 (b)(c)), TV methods are having difficulty since the height of the missing domain is larger than the width of the white stripe. As shown in Figure 6 (d), we get a much better recovery result with

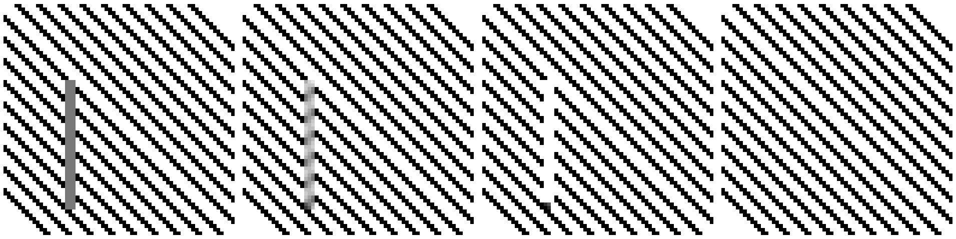


(a) corrupted image (b) model (3), $\lambda = 1$ (c) model (3), $\lambda = 10^3$ (d) with directional constraints

Figure 6. Inpaint a missing block of a vertical stripe, 64 × 64

the direction information. Specifically, the parameters in model (6) are h =DB-2, $\lambda = 1, \mu = 0, \beta = 0.05$. The recovery takes about 0.5 second.

The second experiment (Figure 7) is to remove a thin block from slanted stripes. Figure 7 (b)(c) show unsatisfactory results without the direction information. With direction information, this experiment echos the illustrative example in Figure 2. As expected, we obtain a perfect reconstruction using the proposed model (6). The parameters are h =DB-3, $\lambda = 10^{-2}, \mu = 0, \beta = 20$. The experiment takes about 1.4 seconds.



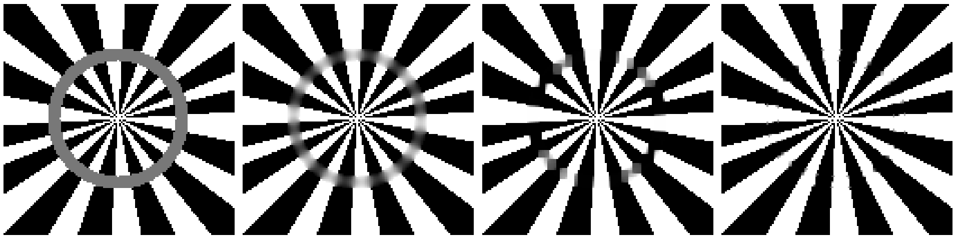
(a) corrupted image (b) model (3), $\lambda = 10$ (c) model (3), $\lambda = 0.01$ (d) with directional constraints

Figure 7. Inpainting a thin missing block in an image with repetitive pattern of slanted stripes, 64 × 64

The third experiment (Figure 8) is to remove an annulus from a sectional image. Figure 8 (b)(c) display typical recovery results with TV. On the contrast, with directional information (parameters are h =DB-1, $\lambda = 1, \mu = 0, \beta = 100$), Figure 8 (d) recovers all the sectional stripes from different directions. This experiment takes about 4 seconds running time.

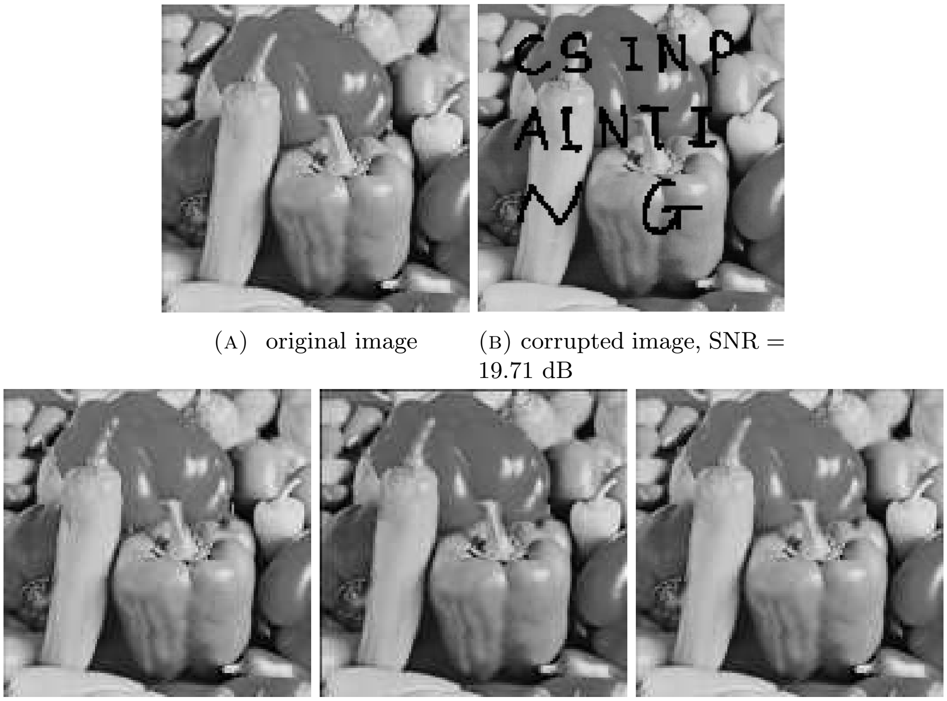
The fourth experiment (Figure 9) is to inpaint a natural image with a text mask. Figure 9 (a)(b) show the original and texted images respectively. Figure 9(c) uses model (5) in a wavelet basis which has no directional constraints. Figure 9(e) inpaints via wavelet sparsity with directional constraints (model (6) with $\lambda = 0$), resulting a better recovery, both visually, and in terms of SNR. We also add the TV result in Figure 9(d) as a comparison and show that the wavelet is doing better in this natural image example.

The fifth experiment (Figure 10) recovers smooth stripes that are masked with 9 thick rectangular blocks. It is also performed with the without TV. Figure 10(c)



(A) corrupted image (B) model (3), $\lambda = 10^3$ (C) model (3), $\lambda = 1$ (D) with directional constraints

Figure 8. Inpaint a missing annulus of a sectional image, 128 × 128



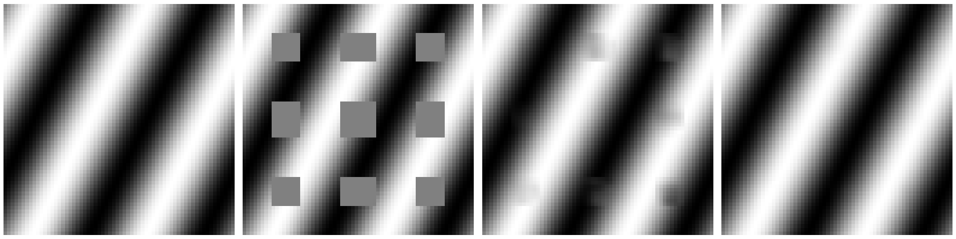
(A) original image (B) corrupted image, SNR = 19.71 dB

(C) model (5), SNR = 26.75 dB (D) TV with directional constraints $\lambda = 10^3$, SNR = 29.05 dB (E) wavelet with directional constraints, SNR=29.20dB

Figure 9. Text removal of Pepper, 128 × 128

uses sparse recovery in DB-4 basis with 5 levels of decomposition. Figure 10(d) adds directional constraints implemented using filters DB-2 and DB-4, which results a visually perfect inpainted image. It is worth pointing out that wavelet sparsity has a clear advantage over TV in this smooth image.

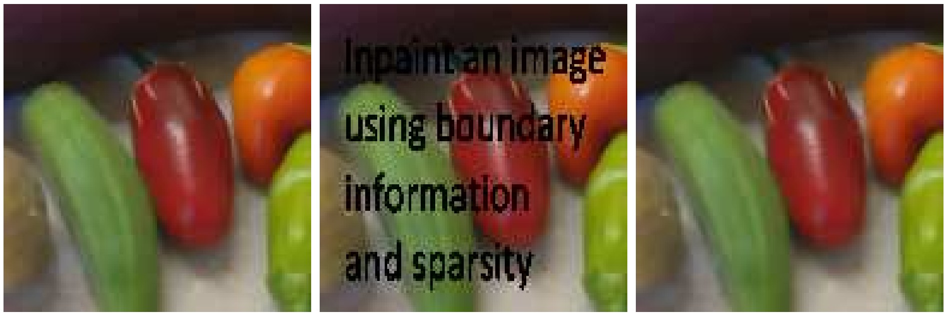
The last experiment (Figure 11) is to remove texts from an RGB image. Red, green and blue channels were processed independently. Using the same indicator of the missing domain, the directional constraints were formed for each of the color channels separately, using filters 'DB-2' and 'DB-4'. All directions with sufficient



(A) original image (B) corrupted image (C) model (5), SNR = 25.87dB (D) with directional constraints, SNR = 44.84dB

FIGURE 10. Smooth stripes inpainting, 64×64

smoothness were included, only centered equations used. $\sqrt{\beta} = 3/4$. Here F is the matrix of the two dimensional 'DB4' transform with 5 levels of decomposition. The SNR for R, G, and B are 35.06, 32.52 and 28.10 dB respectively. The overall SNR is 32.27dB. In this example, we see that our model blends channels well.



(A) original image (B) corrupted image (C) wavelet, with directional constraints, SNR = 32.27dB

FIGURE 11. Text removal of a colored image, 128×128

6. **Conclusions and directions of future work.** We described a method allowing to amend the general scenario of using sparse recovery for inpainting purposes by an efficient use of adaptive one-dimensional directional "sensing" into the unknown domain.

The introduced framework is general enough to offer a lot of flexibility and be successfully utilized in a multitude of image recover scenarios. We would like to outline our further research directions and indicate the avenues of investigation we consider of most importance.

'DB' (Daubechies) wavelet filters were used in this framework since they are the shortest known wavelet filters orthogonal to polynomial changes of the respective order, so our motivation for their use was to use smooth transition constrains for domains of a variety of sizes. The next step will be to investigate the filters of a certain order of length enforcing smoothness of a given order.

While for short filters checking the 8 possible directions of change is adequate, in case of the need to propagate the information into unknown domains along smooth curves, different trajectories of the directional sensing should be considered, depending on the problem context. While not the most efficient technique for the most general recovery, our method will still provide fast and accurate reconstruction given additional geometric assumptions on the data, that we hope to obtain from the preliminary analysis of the unknown domains.

To take it to the next level of complexity, we would like to investigate the relationship between the sparse representation system F and the filter used to enforce the smoothness, and pick the pairings that potentially have the best sparse recovery properties for certain wide classes of missing domains.

Finally, we would like to fine-tune the adaptive basis recovery algorithms such as (6) to fit the inpainting scenario in the patch-wise case, including both the shape of the unknown domain and the proposed smoothness of transitions as the constraints.

Acknowledgments. We are grateful to Dr. W. Czaja and Dr. B. Manning for their help at the early stages of this project. We would like to thank the anonymous reviewers for their helpful comments. This work is supported by NSF DMS-1908880 and NSF DMS-2050028.

REFERENCES

[1] A. Aldroubi, X. Chen and A. M. Powell, Perturbations of measurement matrices and dictionaries in compressed sensing, *Appl. Comput. Harmon. Anal.*, **33** (2012), 282–291.

[2] C. Bao, H. Ji and Z. Shen, Convergence analysis for iterative data-driven tight frame construction scheme, *Appl. Comput. Harmon. Anal.*, **38** (2015), 510–523.

[3] M. Bertalmio, A. L. Bertozzi and G. Sapiro, Navier-Stokes, fluid dynamics, and image and video inpainting, *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, HI, 2001, 355–362.

[4] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, Image inpainting, *Proc. 27th Conference on Computer Graphics and Interactive Techniques*, 2000, 417–424.

[5] M. Bertalmio, L. Vese, G. Sapiro and S. Osher, Simultaneous structure and texture image inpainting, *IEEE Trans. Image Processing*, **12** (2003), 882–889.

[6] A. Bertozzi, S. Esedoḡlu and A. Gillette, Analysis of a two-scale Cahn-Hilliard model for binary image inpainting, *Multiscale Model. Simul.*, **6** (2007), 913–936.

[7] J.-F. Cai, R. H. Chan and Z. Shen, A framelet-based image inpainting algorithm, *Appl. Comput. Harmon. Anal.*, **24** (2008), 131–149.

[8] J.-F. Cai, H. Ji, Z. Shen and G.-B. Ye, Data-driven tight frame construction and image denoising, *Appl. Comput. Harmon. Anal.*, **37** (2014), 89–105.

[9] J.-F. Cai, S. Osher and Z. Shen, Split Bregman methods and frame based image restoration, *Multiscale Model. Simul.*, **8** (2009/10), 337–369.

[10] E. J. Candès, Y. C. Eldar, D. Needell and P. Randall, Compressed sensing with coherent and redundant dictionaries, *Appl. Comput. Harmon. Anal.*, **31** (2011), 59–73.

[11] E. J. Candès, X. Li, Y. Ma and J. Wright, Robust principal component analysis?, *J. ACM*, **58** (2011), 37pp.

[12] E. J. Candès, J. K. Romberg and T. Tao, Stable signal recovery from incomplete and inaccurate measurements, *Comm. Pure Appl. Math.*, **59** (2006), 1207–1223.

[13] P. G. Casazza, The art of frame theory, *Taiwanese J. Math.*, **4** (2000), 129–201.

[14] T. F. Chan and J. Shen, Mathematical models for local nontexture inpaintings, *SIAM J. Appl. Math.*, **62** (2001/02), 1019–1043.

[15] A. Criminisi, P. Pérez and K. Toyama, Region filling and object removal by exemplar-based image inpainting, *IEEE Trans. Image Processing*, **13** (2004), 1200–1212.

[16] J. Darbon and M. Sigelle, A fast and exact algorithm for total variation minimization, in *Pattern Recognition and Image Analysis*, Lecture Notes in Computer Science, 3522, Springer, 2005, 351–359.

[17] J. Dobrosotskaya and W. Guo, Data adaptive multi-scale representations for image analysis, in *Wavelets and Sparsity XVIII*, 11138, International Society for Optics and Photonics, 2019.

[18] B. Dong and Z. Shen, Image restoration: A data-driven perspective, *Proceedings of the 8th International Congress on Industrial and Applied Mathematics*, Higher Ed. Press, Beijing, 2015, 65–108.

[19] D. L. Donoho, Compressed sensing, *IEEE Trans. Inform. Theory*, **52** (2006), 1289–1306.

[20] A. A. Efros and T. K. Leung, Texture synthesis by non-parametric sampling, *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, 1999.

[21] M. Elad, J.-L. Starck, P. Querre and D.-L. Donoho, Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA), *Appl. Comput. Harmon. Anal.*, **19** (2005), 340–358.

[22] B. Han, G. Kutyniok and Z. Shen, Adaptive multiresolution analysis structures and shearlet systems, *SIAM J. Numer. Anal.*, **49** (2011), 1921–1946.

[23] E. J. King, G. Kutyniok and X. Zhuang, Analysis of inpainting via clustered sparsity and microlocal analysis, *J. Math. Imaging Vision*, **48** (2014), 205–234.

[24] M. Lustig, D. Donoho and J. M. Pauly, Sparse MRI: The application of compressed sensing for rapid MR imaging, *Magnetic Resonance in Medicine*, **58** (2007), 1182–1195.

[25] S. Mallat, *A Wavelet Tour of Signal Processing*, Elsevier/Academic Press, Amsterdam, 2009.

[26] Y. Meyer, *Wavelets and Operators*, Cambridge Studies in Advanced Mathematics, 37, Cambridge University Press, Cambridge, 1992.

[27] N. Parikh and S. Boyd, *Proximal Algorithms*, Now Foundations and Trends, 2014, 128pp.

[28] Y. Quan, H. Ji and Z. Shen, Data-driven multi-scale non-local wavelet frame construction and image recovery, *J. Sci. Comput.*, **63** (2015), 307–329.

[29] L. I. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Phys. D*, **60** (1992), 259–268.

[30] S. F. D. Waldron, *An Introduction to Finite Tight Frames*, Applied and Numerical Harmonic Analysis, Birkhäuser/Springer, New York, 2018.

[31] 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), 248–272.

[32] Z. Xu and J. Sun, Image inpainting by patch propagation using patch sparsity, *IEEE Trans. Image Process.*, **19** (2010), 1153–1165.

Received February 2020; 1st revision June 2020; 2nd revision October 2020.

E-mail address: chenxuemei@uncw.edu
E-mail address: jxd365@case.edu