

THE IMAGE DEBLURRING PROBLEM: MATRICES, WAVELETS, AND MULTILEVEL METHODS

DAVID AUSTIN¹, MALENA I. ESPAÑOL², AND MIRJETA PASHA³

1. INTRODUCTION

After the launch of the Hubble Space Telescope in 1990, astronomers were gravely disappointed by the quality of the images as they began to arrive. Due to miscalibrated testing equipment, the telescope's primary mirror had been ground to a shape that differed slightly from the one intended, resulting in the misdirection of incoming light as it moved through the optical system. The blurry images did little to justify the telescope's \$1.5 billion price tag [1]. Three years later, space shuttle astronauts installed a specially designed corrective optical system that essentially fixed the problem and yielded spectacular images (see Figure 1). In the meantime, mathematicians devised several ways to convert the blurry images into high-quality images. The process of mathematical deblurring is the focus of this article.



FIGURE 1. Hubble's view of the M100 galaxy, soon after launch on the left and after corrective optics were installed in 1993. NASA / ESA

Many factors can cause an image to become blurry, such as the movement of the imaging device or the target object, focus errors, or the presence of atmospheric turbulence [21]. Indeed, the need for image deblurring goes beyond Hubble's story. For instance, the image deblurring problem is encountered in many applications such as pattern recognition, computer vision, and machine intelligence. Moreover, image deblurring shares the same mathematical formulation as other imaging modalities. For instance, in many cases we have only limited opportunities to capture an image; this is particularly true of medical images, such as computerized tomography (CT), proton computed tomography (pCT), and magnetic resonance imaging (MRI), for which equipment and patient

availability are scarce resources. In cases such as these, we need a way to extract meaningful information from noisy images that have been imperfectly collected.

In this article, we will describe a mathematical model of how digital images become blurry as well as several mathematical issues that arise when we try to undo the blurring. While blurring may be effectively modeled by a linear process, we will see that deblurring is not as simple as inverting that linear process. Indeed, deblurring belongs to an important class of problems known as *discrete ill-posed problems* [19], and we will introduce some techniques that have become standard for solving them. In addition, we will describe some matrix structures in the linear operators that will allow more efficient computations.

2. DIGITAL IMAGES AND BLURRING

We start by describing digital images and a process by which they become blurry. As illustrated in Figure 2, the lens of a digital camera directs photons entering the camera onto a charge-coupled device (CCD), which consists of a rectangular $p \times q$ array of detectors. Each detector in the CCD converts a count of the photons into an electrical signal that is digitized by an analog-to-digital converter (ADC). The result is a digital image stored as a $p \times q$ matrix whose entries represent the intensities of light recorded by each of the CCD's detectors.

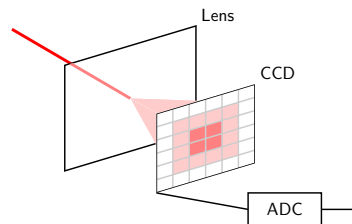


FIGURE 2. A simple model of how a digital image is created.

A grayscale image is represented by a single matrix with integer entries describing the brightness at each location. A color image is represented by three matrices that describe the colors in terms of their red, green, and blue constituents. Of course, we may see these matrices by

zooming in on a digital image until we see individual pixels, picture elements that contain only one intensity value.

Perhaps due to imperfections in the camera's lens or the lens being improperly focused, it is inevitable that photons intended for one pixel bleed over into adjacent pixels, and this leads to blurring of the image. To illustrate, we will consider grayscale images comprised of arrays of 64×64 pixels. In Figure 3, the image on the left shows a single pixel illuminated while on the right we see how photons intended for this single pixel have spilled over into adjacent pixels to create a blurred image.

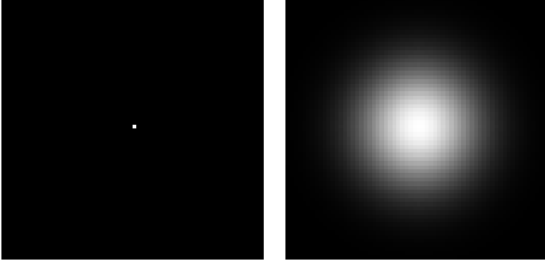


FIGURE 3. The intensity from a single pixel, shown on the left, is spread out across adjacent pixels according to a Gaussian blur, as seen on the right.

There are several models used to describe blurring. A simple one that we choose here has the light intensity contained in a single pixel (i, j) spilling over into an adjacent pixel (k, l) according to the Gaussian point spread function

$$\begin{aligned} & \frac{1}{N} \exp \left(-\frac{1}{2} \left(\frac{k-i}{s} \right)^2 - \frac{1}{2} \left(\frac{l-j}{s} \right)^2 \right) \\ &= \frac{1}{N} \exp \left(-\frac{1}{2} \left(\frac{k-i}{s} \right)^2 \right) \exp \left(-\frac{1}{2} \left(\frac{l-j}{s} \right)^2 \right), \end{aligned} \quad (1)$$

where s is a parameter that controls the spread in the intensity and N is a normalization constant so that the total intensity sums to 1. As we will see later, the fact that the two-dimensional Gaussian can be written as a product of one-dimensional Gaussians has important consequences for our ability to efficiently represent the blurring process as a linear operator.

Though we visually experience a grayscale image as a $p \times q$ matrix of pixels \mathbf{X} , we will mathematically represent an image as a pq -dimensional vector \mathbf{x} by stacking the columns of \mathbf{X} on top of one another. That is, $\mathbf{x} = \text{vec}(\mathbf{X}) = (\mathbf{X}_{11}, \dots, \mathbf{X}_{p1}, \mathbf{X}_{12}, \dots, \mathbf{X}_{p2}, \dots, \mathbf{X}_{1q}, \dots, \mathbf{X}_{pq})^T$ with \mathbf{X}_{ij} being the intensity value of the pixel at row i and column j . The blurring process is linear as the number of photons that arrive at one pixel is the sum of

the number of misdirected photons intended for nearby pixels. Therefore, if \mathbf{B} is the blurred image represented by the vector $\mathbf{b} = \text{vec}(\mathbf{B})$, then there is a blurring matrix \mathbf{A} that blurs the original image \mathbf{x} into the blurred image $\mathbf{b} = \mathbf{A}\mathbf{x}$. Furthermore, we assume that both images \mathbf{X} and \mathbf{B} have the same size, as do \mathbf{x} and \mathbf{b} , which implies that the matrix \mathbf{A} is square. *Deblurring* refers to the inverse process of recovering the original image \mathbf{x} from its blurred counterpart \mathbf{b} and includes situations, unlike the one we describe here, where \mathbf{A} is not invertible or even square.

Each column of \mathbf{A} is the result of blurring a single pixel, which means that each column of \mathbf{A} represents the point spread function centered at its corresponding pixel (i, j) . When that center is near the edge of the image, some photons will necessarily be lost outside the image, and there are a few options for how to incorporate this fact into our model. In real-life settings, it is possible to have knowledge only over a finite region, the so called Field of View (FOV), that defines the range that a user can see from an object. It is then necessary to make an assumption on what is outside the FOV by means of the boundary conditions. For instance, one option to overcome the loss outside the FOV is to simply accept that loss, in which case we say that the matrix has zero boundary conditions. This has the effect of assuming that the image is black (pixel values are zero) outside the FOV, which can lead to an artificial black border around a deblurred image (see the image on the left on Figure 4).

In some contexts, it can be advantageous to impose reflexive boundary conditions, which assumes that the photons are reflected back onto the image. In other scenarios of interest, periodic boundary conditions, which assumes the lost photons reappear on the opposite side of the image as if the image repeats itself indefinitely in all directions outside the FOV, are a suitable fit. Nevertheless, in practical settings, we periodically extend only some pixel values close to the boundary (see the image on the right on Figure 4).



FIGURE 4. Image with assumed zero boundary conditions is shown on the left and one with assumed periodic boundary condition is shown on the right. The red box represents the FOV.

3. ADDING NOISE

Let us consider the grayscale image \mathbf{x}^{true} shown on the left of Figure 5 and its blurred version $\mathbf{b}^{\text{true}} = \mathbf{A}\mathbf{x}^{\text{true}}$ shown on the right.

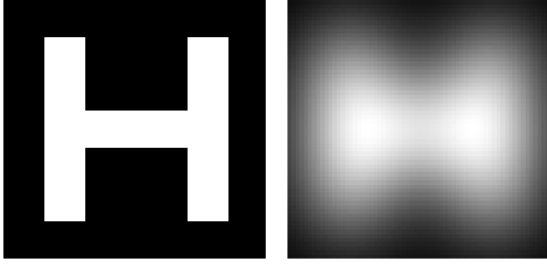


FIGURE 5. An image \mathbf{x}^{true} on the left is blurred to obtain \mathbf{b}^{true} on the right.

If we had access to \mathbf{b}^{true} , it would be easy enough to recover \mathbf{x}^{true} by simply solving the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}^{\text{true}}$ since our model gives a square, invertible matrix \mathbf{A} . However, the conversion of photon counts into an electrical signal by the CCD and then a digital reading by the ADC introduces electrical noise into the image that is ultimately recorded. The recorded image \mathbf{b} is therefore a noisy approximation to the true blurred image \mathbf{b}^{true} , so we write $\mathbf{b} = \mathbf{b}^{\text{true}} + \mathbf{e}$, where \mathbf{e} is a noise vector whose entries are independent and identically distributed random variables. There are various models used to describe the kind of noise added. For instance, we might assume that the noise is Gaussian white noise, which means that the entries in \mathbf{e} are sampled from a normal distribution with mean zero. In our example, we assume that $\|\mathbf{e}\|_2 = 0.001\|\mathbf{b}^{\text{true}}\|_2$, that is, the white noise is about 0.1% of the image \mathbf{b}^{true} . As seen in Figure 6, this level of noise cannot be easily detected.

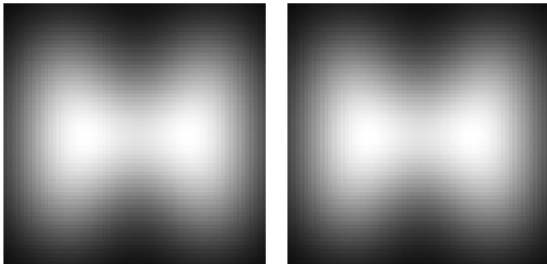


FIGURE 6. The blurred image \mathbf{b}^{true} on the left with a small amount of Gaussian white noise added to obtain \mathbf{b} on the right.

Two other models of noise are illustrated in Figure 7. The image on the left contains what is called Poisson noise. Under low light conditions, as encountered in astronomical imaging, the number of photons that arrive on the CCD are very low and therefore the noise produced is better modeled by a Poisson distribution than a Gaussian one [33]. If the digital image is transmitted over a communication channel, some of the transmitted bits may be corrupted in transmission, resulting in “salt and pepper” noise, the effect of which is shown in the image to the right [21]. This noise can be modeled by a uniform distribution whose standard deviation is inversely proportional to the number of bits used. In the rest of the article, we will assume that we have Gaussian white noise. Some references on how to model and solve other noise models are included in [35, 4, 10, 24].

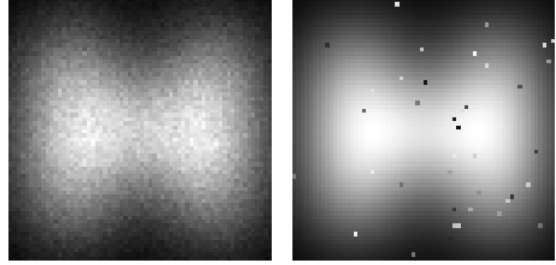


FIGURE 7. Poisson noise is added to the blurred image on the left and salt and pepper noise on the right.

Because our recorded image \mathbf{b} is a good approximation of \mathbf{b}^{true} , we might naïvely expect to find a good approximation of \mathbf{x}^{true} by solving the linear system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$. However, its solution, which we call \mathbf{x}^{LS} , turns out to be very different from the original image \mathbf{x}^{true} as is seen in Figure 8. As we will soon see, this behavior results from the fact that deblurring is a discrete linear ill-posed problem.

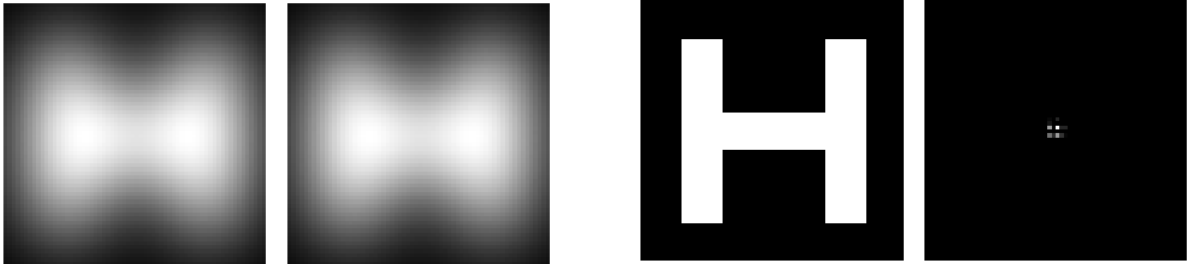


FIGURE 8. On the left we see the original image \mathbf{x}^{true} while the right shows \mathbf{x}^{LS} , the solution to the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$, where \mathbf{b} is the noisy recorded image.

We are now faced with two questions: how can we reconstruct the original image \mathbf{x}^{true} more faithfully and how can we do it in a computationally efficient way. For instance, today's typical phone photos have around 10 million pixels, which means that the number of entries in the blurring matrix \mathbf{A} is about 100 trillion. Working with a matrix of that size will require some careful thought.

4. DISCRETE LINEAR ILL-POSED PROBLEMS

The singular value decomposition (SVD) of the matrix \mathbf{A} offers insight into why the naively reconstructed image \mathbf{x}^{LS} differs so greatly from the original \mathbf{x}^{true} . Consider both vectors \mathbf{x}^{true} and \mathbf{b} of dimension $m = pq$. Now suppose that $\mathbf{A} \in \mathbb{R}^{m \times m}$ has full rank and its SVD is given by

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{\ell=1}^m \sigma_{\ell} \mathbf{u}_{\ell} \mathbf{v}_{\ell}^T,$$

where $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_m) \in \mathbb{R}^{m \times m}$ and $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_m) \in \mathbb{R}^{m \times m}$ are matrices having orthonormal columns so that $\mathbf{U}^T \mathbf{U} = \mathbf{V}^T \mathbf{V} = \mathbf{I}$, and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_m)$, with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m > 0$, that is to say, $\mathbf{\Sigma}$ is an $m \times m$ diagonal matrix, whose entries are positive and appear in decreasing order. The scalars σ_{ℓ} are the *singular values* of \mathbf{A} and the vectors \mathbf{u}_{ℓ} and \mathbf{v}_{ℓ} are the *left* and *right singular vectors* of \mathbf{A} , respectively. In particular, the singular value decomposition provides orthonormal bases defined by the columns of \mathbf{U} and \mathbf{V} , for the range and domain spaces, respectively, so that $\mathbf{A}\mathbf{v}_{\ell} = \sigma_{\ell} \mathbf{u}_{\ell}$. Notice that we are assuming that the matrix \mathbf{A} is square and full rank, since all the singular values are positive, and therefore invertible. This is natural in the image deblurring model described here. In other inverse problems, however, the matrix \mathbf{A} may be square but rank-deficient or even rectangular, situations whose solutions require more detailed considerations [19].

Discrete linear ill-posed problems are characterized by three properties, which are satisfied by the deblurring problem. First, the singular values σ_{ℓ} decrease to zero without a large gap separating a group of large singular values from a group of smaller ones. The plot of the singular values in Figure 9 shows that the difference between the largest and smallest singular values is about thirteen orders of magnitude, which indicates that \mathbf{A} is highly ill-conditioned.

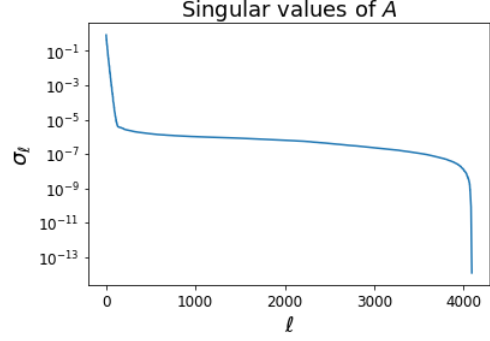


FIGURE 9. The singular values σ_{ℓ} of \mathbf{A} .

Second, the singular vectors \mathbf{u}_{ℓ} and \mathbf{v}_{ℓ} become more and more oscillatory as ℓ increases. Figure 10 shows images \mathbf{V}_{ℓ} representing eight right singular vectors \mathbf{v}_{ℓ} ($\mathbf{v}_{\ell} = \text{vec}(\mathbf{V}_{\ell})$) of the blurring matrix constructed above and demonstrates how the frequency of the oscillations increases as ℓ increases. Since the blurring matrix \mathbf{A} spreads out any peaks in an image, it tends to dampen high frequency oscillations. Therefore, a right singular vector \mathbf{v}_{ℓ} representing a high frequency will correspond to a small singular value σ_{ℓ} since $\mathbf{A}\mathbf{v}_{\ell} = \sigma_{\ell} \mathbf{u}_{\ell}$. This oscillatory behavior of the singular vectors and their relation to the singular values appears more generally with other blurring operators defined by other kernels and in other discrete ill-posed inverse problems due to the underlying continuous model [19].

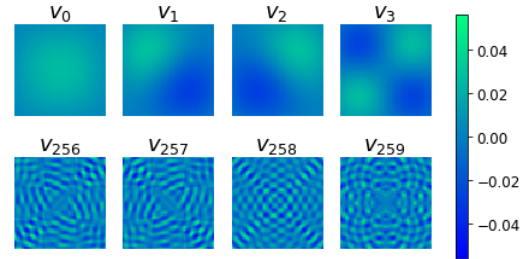


FIGURE 10. Images corresponding to eight right singular vectors \mathbf{v}_{ℓ} .

The third property of discrete linear ill-posed problems is known as the *discrete Picard condition*, which says that the coefficients of \mathbf{b}^{true} expressed in the left singular basis, $|\mathbf{u}_{\ell}^T \mathbf{b}^{\text{true}}|$, on average, decay to zero faster than the singular values σ_{ℓ} [19]. This is illustrated on the left of Figure 11, which shows both $|\mathbf{u}_{\ell}^T \mathbf{b}^{\text{true}}|$ and the singular values σ_{ℓ} . Since $\mathbf{u}_{\ell}^T \mathbf{b}^{\text{true}} = \sigma_{\ell} \mathbf{v}_{\ell}^T \mathbf{x}^{\text{true}}$, the discrete Picard condition holds when the original image \mathbf{x}^{true} is not dominated by high frequency contributions $\mathbf{v}_{\ell}^T \mathbf{x}^{\text{true}}$ for large ℓ . This is a reasonable assumption with most digital photographs [21].

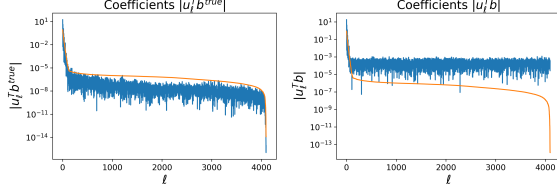


FIGURE 11. The coefficients $|\mathbf{u}_\ell^T \mathbf{b}^{\text{true}}|$ are seen on the left while the coefficients $|\mathbf{u}_\ell^T \mathbf{b}|$ are on the right.

By contrast, the added white noise causes the coefficients $|\mathbf{u}_\ell^T \mathbf{b}| = |\mathbf{u}_\ell^T \mathbf{b}^{\text{true}} + \mathbf{u}_\ell^T \mathbf{e}|$ to remain relatively constant for large ℓ , as can be seen on the right of Figure 11. However, both plots agree for small values of ℓ .

Writing both vectors \mathbf{x}^{LS} and \mathbf{x}^{true} as a linear combination of the right singular vectors \mathbf{v}_ℓ , we see that

$$\mathbf{x}^{\text{true}} = \sum_{\ell=1}^m \frac{\mathbf{u}_\ell^T \mathbf{b}^{\text{true}}}{\sigma_\ell} \mathbf{v}_\ell$$

and

$$\begin{aligned} \mathbf{x}^{LS} &= \sum_{\ell=1}^m \frac{\mathbf{u}_\ell^T \mathbf{b}}{\sigma_\ell} \mathbf{v}_\ell = \sum_{\ell=1}^m \frac{\mathbf{u}_\ell^T (\mathbf{b}^{\text{true}} + \mathbf{e})}{\sigma_\ell} \mathbf{v}_\ell \\ &= \sum_{\ell=1}^m \frac{\mathbf{u}_\ell^T \mathbf{b}^{\text{true}}}{\sigma_\ell} \mathbf{v}_\ell + \sum_{\ell=1}^m \frac{\mathbf{u}_\ell^T \mathbf{e}}{\sigma_\ell} \mathbf{v}_\ell \\ &= \mathbf{x}^{\text{true}} + \sum_{\ell=1}^m \frac{\mathbf{u}_\ell^T \mathbf{e}}{\sigma_\ell} \mathbf{v}_\ell. \end{aligned}$$

Because the singular values approach zero, the coefficients $|\mathbf{u}_\ell^T \mathbf{e}|/\sigma_\ell$ grow extremely large, as seen in Figure 12. Therefore, \mathbf{x}^{LS} includes a huge contribution from high-frequency right singular vectors \mathbf{v}_ℓ , which means that \mathbf{x}^{LS} is very oscillatory and not at all related to the original image \mathbf{x}^{true} that we seek to reconstruct.

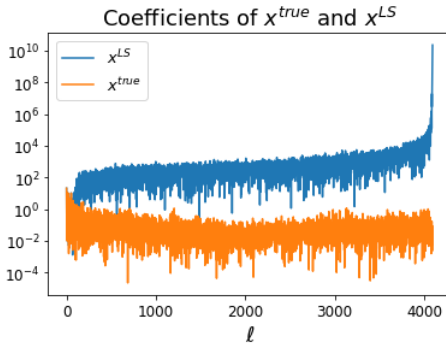


FIGURE 12. The coefficients of \mathbf{x}^{true} and \mathbf{x}^{LS} .

We also note here that the size of the coefficients shown in Figure 12 cause the norm $\|\mathbf{x}^{LS}\|_2$ to be extremely large. As we will see shortly, we will consider this norm as a

measure of the amount of noise in the reconstructed image.

5. REGULARIZATION

As an alternative to accepting \mathbf{x}^{LS} as our reconstructed image, we instead approximate \mathbf{x}^{true} by filtering out the contributions from the high frequency singular vectors, which are magnified through division by small singular values, while retaining as much information as possible from the measured data \mathbf{b} . While we know the noise vector \mathbf{e} in the example presented here, in practice we do not, which means we cannot expect to get rid of the noise completely. Therefore, the goal is to design filtering methods that dampen the effects of the noise given that very little is known about it. This process is known as *regularization*, and there are several possible approaches we can follow.

A first natural idea is to introduce a set of *filtering factors* ϕ_ℓ on the SVD expansion and construct a regularized solution as

$$\mathbf{x}^{\text{reg}} = \sum_{\ell=1}^m \phi_\ell \frac{\mathbf{u}_\ell^T \mathbf{b}}{\sigma_\ell} \mathbf{v}_\ell,$$

with the filter factors being $\phi_\ell \approx 0$ for large values of ℓ when the noise dominates and $\phi_\ell \approx 1$ for small values of ℓ , which are the terms in the expansion where the components of \mathbf{b} and \mathbf{b}^{true} are closest.

One option is to define $\phi_\ell = 1$ for ℓ smaller than some cut-off $k < m$ and $\phi_\ell = 0$ otherwise. That is, we can simply truncate the expansion of \mathbf{x}^{LS} in terms of right singular vectors in an attempt to minimize the contribution from the terms $|\mathbf{u}_\ell^T \mathbf{e}|/\sigma_\ell$ for large ℓ . Then, the obtained regularized solution would be

$$\mathbf{x}^{\text{TSVD}} = \sum_{\ell=1}^k \frac{\mathbf{u}_\ell^T \mathbf{b}}{\sigma_\ell} \mathbf{v}_\ell.$$

This solution is known as the truncated SVD (TSVD).

Remember, however, that the singular values in discrete ill-posed problems approach zero without having a gap that would form a natural cut-off point. Instead, Tikhonov regularization chooses the filtering factors

$$\phi_\ell = \frac{\sigma_\ell^2}{\sigma_\ell^2 + \lambda^2}$$

for some parameter λ whose choice will be discussed later. For now, notice that $\phi_\ell \approx 1$ when $\sigma_\ell \gg \lambda$ and $\phi_\ell \approx 0$ when $\sigma_\ell \ll \lambda$. This has the effect of truncating the singular vector expansion at the point where the singular values pass through λ but doing so more smoothly. Thus, the Tikhonov solution can be written as

$$\mathbf{x}^{\text{Tik}} = \sum_{\ell=1}^m \frac{\sigma_\ell}{\sigma_\ell^2 + \lambda^2} (\mathbf{u}_\ell^T \mathbf{b}) \mathbf{v}_\ell,$$

which may also be rewritten as

$$\mathbf{x}^{\text{Tik}} = (\mathbf{A}^T \mathbf{A} + \lambda^2 \mathbf{I})^{-1} \mathbf{A}^T \mathbf{b}.$$

This demonstrates that \mathbf{x}^{Tik} solves the least squares problem

$$\mathbf{x}^{\text{Tik}} = \underset{\mathbf{x}}{\operatorname{argmin}} \{ \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda^2 \|\mathbf{x}\|_2^2 \}, \quad (2)$$

where $\operatorname{argmin}_{\mathbf{x}} \{f(\mathbf{x})\}$ returns the value of \mathbf{x} which minimizes the function $f(\mathbf{x})$.

Equation (2) is a helpful reformulation of Tikhonov regularization. First, writing \mathbf{x}^{Tik} as the solution of a least squares problem provides us with efficient computational alternatives to finding the SVD of \mathbf{A} [19]. Moreover, the minimization problem provides insight into choosing the optimal value of the regularization parameter λ , as we will soon see.

By the way, this formulation of Tikhonov regularization shows its connection to ridge regression, a data science technique for tuning a linear regression model in the presence of multicollinearity to improve its predictive accuracy.

Let us investigate the meaning of (2). Notice that

$$\|\mathbf{A}\mathbf{x}^{\text{true}} - \mathbf{b}\|_2^2 = \|\mathbf{b}^{\text{true}} - \mathbf{b}\|_2^2 = \|\mathbf{e}\|_2^2, \quad (3)$$

which is relatively small. We therefore consider the residual $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ as a measure of how far away we are from the original image \mathbf{x}^{true} . Remember that $|\mathbf{u}_\ell^T \mathbf{e}|/\sigma_\ell$, the contributions to \mathbf{x}^{LS} from the added noise, cause $\|\mathbf{x}^{LS}\|_2$ to be very large. Consequently, we think of the second term in (2) as measuring the amount of noise in \mathbf{x} .

The regularization parameter λ allows us to balance these two sources of error. For instance, when λ is small, the Tikhonov solution \mathbf{x}^{Tik} , which is the minimum of (2), will have a small residual $\|\mathbf{A}\mathbf{x}^{\text{Tik}} - \mathbf{b}\|_2$ at the expense of a large norm $\|\mathbf{x}^{\text{Tik}}\|_2$. In other words, we tolerate a noisy regularized solution in exchange for a small residual. On the other hand, if λ is large, the Tikhonov solution will have a relatively large residual in exchange for filtering out a lot of the noise.

6. CHOOSING THE REGULARIZING PARAMETER λ

When applying Tikhonov regularization for solving discrete ill-posed inverse problems, a question of high interest is: How can we define the best regularization parameter λ ? A common and well known technique is to create a log-log plot of the residuals $\|\mathbf{A}\mathbf{x}^{\text{Tik}} - \mathbf{b}\|_2$ and the norm $\|\mathbf{x}^{\text{Tik}}\|_2$ as we vary λ . This plot, as shown in Figure 13, is usually called an *L-curve* due to its characteristic shape [22].

As mentioned earlier, small values of λ lead to noisy regularized solutions while large values of λ produce large residuals. This means that we move across the *L-curve* from the upper left to the lower right as we increase λ .

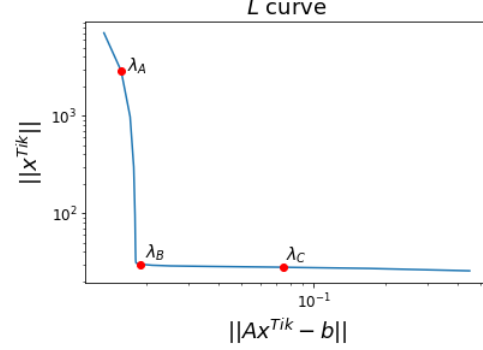


FIGURE 13. The *L-curve* in our sample deblurring problem. The three indicated points correspond to values of $\lambda = \lambda_A, \lambda_B$, and λ_C .

Since the filtering factors satisfy $\phi_i \approx 1$ when $\sigma \gg \lambda$ and $\phi_\ell \approx 0$ when $\sigma_\ell \ll \lambda$, we will view the point where $\sigma_\ell = \lambda$ as an indication of where we begin filtering. Let us first consider the value $\lambda = \lambda_A$, which produces the point on the *L-curve* indicated in Figure 13. The resulting coefficients $|\mathbf{u}_\ell^T \mathbf{b}|$ and $\phi_\ell |\mathbf{u}_\ell^T \mathbf{b}|$ are shown in Figure 14. Filtering begins roughly where the plot of singular values crosses the horizontal line indicating the value of λ_A .

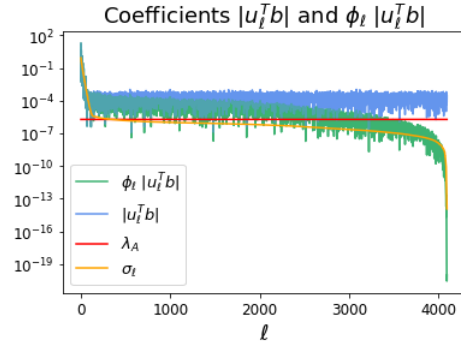


FIGURE 14. The choice $\lambda = \lambda_A$ leads to under-smoothing.

While we have filtered out some of the noise, it appears that there is still a considerable amount of noise present. This is reflected by the position of the corresponding point on the *L-curve* since the norm $\|\mathbf{x}^{\text{Tik}}\|_2$ is relatively large. This choice of λ is too low, and we say that the regularized solution is under-smoothed.

Alternatively, let us consider the regularized solution constructed with $\lambda = \lambda_C$ as indicated on Figure 13. This leads to the coefficients $|\mathbf{u}_\ell^T \mathbf{b}|$ and $\phi_\ell |\mathbf{u}_\ell^T \mathbf{b}|$ shown in Figure 15.

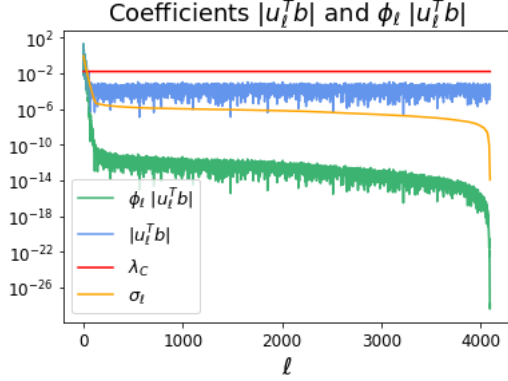


FIGURE 15. The choice $\lambda = \lambda_C$ leads to over-smoothing.

In this case, we begin filtering too soon so that, while we have filtered the noise, we have also discarded some of the information present in \mathbf{b} , which is reflected in the relatively large residual $\|\mathbf{A}\mathbf{x}^{\text{Tik}} - \mathbf{b}\|_2$. This choice of λ is too large, and we say that the Tikhonov solution is over-smoothed.

Finally, considering the case where $\lambda = \lambda_B$ gives the Tikhonov solution that appears at the sharp bend of the L -curve in Figure 13. The resulting coefficients shown in Figure 16 inspire confidence that this is a good choice for λ .

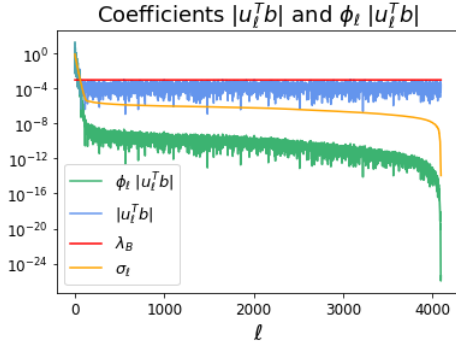


FIGURE 16. The choice $\lambda = \lambda_B$ gives the optimal amount of smoothing.

In this case, decreasing λ causes us to move upward on the L -curve; we are allowing more high frequency components into the solution without improving the residual. Increasing λ causes us to move right on the L -curve; we are losing information as the residual increases without removing relevant high frequency components. Therefore, $\lambda = \lambda_B$ is our optimal value.

Figure 17 shows the image \mathbf{x}^{Tik} obtained using this optimal parameter $\lambda = \lambda_B$. While it is not a perfect reconstruction of the original image \mathbf{x}^{true} , it is a significant improvement over the recorded image \mathbf{b} . Accurately reproducing the sharp boundaries that occur in \mathbf{x}^{true} requires high-frequency contributions that we have necessarily filtered out.

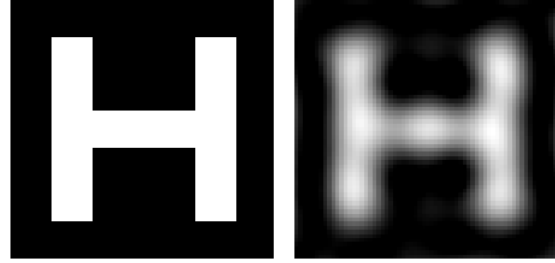


FIGURE 17. The reconstructed image \mathbf{x}^{Tik} using the regularization parameter determined by the L -curve is seen on the right, along with the original image \mathbf{x}^{true} on the left for comparison.

The L -curve furnishes a practical way to identify the optimal regularizing parameter as there are techniques that allow us to identify the point of maximal curvature by computing the Tikhonov solution for just a few choices of λ . However, this technique should not be applied uncritically as there are cases in which the optimal Tikhonov solution does not converge to the true image as the added error approaches zero.

An alternative technique, known as the Discrepancy Principle [15], relies on an estimate of the size of the error $\|\mathbf{e}\|_2$. Remember from (3) that we have $\|\mathbf{A}\mathbf{x}^{\text{true}} - \mathbf{b}\|_2 = \|\mathbf{e}\|_2$. Moreover, the SVD description of \mathbf{x}^{Tik} provides a straightforward explanation for why the residual $\|\mathbf{A}\mathbf{x}^{\text{Tik}} - \mathbf{b}\|_2$ is an increasing function of λ . If we know $\|\mathbf{e}\|_2$, we simply choose the optimal λ to be the one where $\|\mathbf{A}\mathbf{x}^{\text{Tik}} - \mathbf{b}\|_2 = \|\mathbf{e}\|_2$.

Other well-known methods for choosing the parameter λ include the Generalized Cross Validation (GCV) [17], that chooses λ to maximize the accuracy with which we can predict the value of a pixel that has been omitted, the unbiased predictive risk estimator (UPRE) [28], and more recently, methods based on learning when training data are available [11]. Most of these methods can also be used to find the regularization parameter k involved in the TSVD solution.

7. OTHER REGULARIZATION TECHNIQUES

Looking at the variational formulation (2) of Tikhonov regularization, it is easy to see how it can be extended to

define other regularization methods by, for example, using different regularization terms.

7.1. General-form Tikhonov Regularization. The Tikhonov regularization formulation (2) can be generalized to

$$\min_{\mathbf{x}} \{\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{Lx}\|_2^2\}, \quad (4)$$

by incorporating a matrix \mathbf{L} , which is called the regularization matrix [19]. Its choice is problem dependent and can significantly affect the quality of the reconstructed solution. Several choices of the regularization matrix involve discretization of the derivative operators or framelet [3, 5, 6, 7, 23] and wavelet transformations depending on the application. The only requirement on \mathbf{L} is that it should satisfy

$$\mathcal{N}(\mathbf{A}) \cap \mathcal{N}(\mathbf{L}) = \{0\},$$

where $\mathcal{N}(\mathbf{M})$ denotes the null space of the matrix \mathbf{M} . The general Tikhonov minimization problem (4) has the unique solution

$$\mathbf{x}^{\text{GTik}} = (\mathbf{A}^T \mathbf{A} + \lambda \mathbf{L}^T \mathbf{L})^{-1} \mathbf{A}^T \mathbf{b}$$

for any $\lambda > 0$.

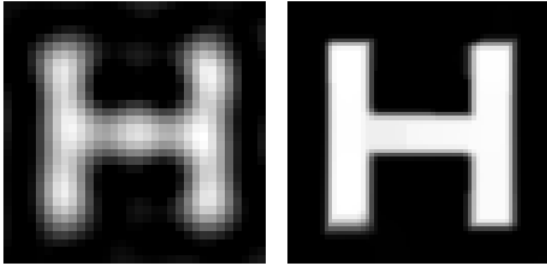


FIGURE 18. The reconstructed images using the optimal regularization parameter and the discretization of the first derivative operator with the 2-norm regularization on the left, and TV regularization on the right.

In Figure 18, we reconstruct the image by applying the discretization of the two-dimensional first derivative operator for zero boundary conditions, that is, the matrix \mathbf{L} takes the form of

$$\mathbf{L} = \begin{pmatrix} \mathbf{I} & \otimes & \mathbf{L}_1 \\ \mathbf{L}_1 & \otimes & \mathbf{I} \end{pmatrix} \text{ with } \mathbf{L}_1 = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix} \quad (5)$$

where \otimes is the *Kronecker product* [18] defined, for matrices \mathbf{B} and \mathbf{C} , by

$$\mathbf{B} \otimes \mathbf{C} = \begin{pmatrix} b_{11}\mathbf{C} & b_{12}\mathbf{C} & \dots & b_{1m}\mathbf{C} \\ b_{21}\mathbf{C} & b_{22}\mathbf{C} & \dots & b_{2m}\mathbf{C} \\ \vdots & \vdots & & \vdots \\ b_{m1}\mathbf{C} & b_{m2}\mathbf{C} & \dots & b_{mm}\mathbf{C} \end{pmatrix}.$$

7.2. Total Variation Regularization. In many applications, the image to be reconstructed is known to be piecewise constant with regular and sharp edges, like the one we are using as an example in this article. Total variation (TV) regularization is a popular choice that allows the solution to preserve edges [30]. Such regularization can be formulated as

$$\min_{\mathbf{x}} \{\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{Lx}\|_1\},$$

where \mathbf{L} is the anisotropic TV operator, which once discretized, is the same as (5). Looking at Figure 18, we can see that even though we are using the same operator \mathbf{L} , the norms used in the regularization terms are different and that makes a huge difference. But there is a higher cost to finding the TV solution due to the fact that the minimization functional is not differentiable. Still, there are many algorithms to find its minimum. Here, we apply the Iteratively Reweighted Least Squares (IRLS), which solves a sequence of general-form Tikhonov problems [29]. So, instead of solving only one Tikhonov problem, we solve many.

The TV approach is also very commonly used in compressed sensing where the signal to be reconstructed is sparse in its original domain or in some transformed domain [8].

Recently it has been used in the context of regularizing large-scale dynamic inverse problems [27] as well as in learning when training data are available [2].

8. MATRIX STRUCTURES

8.1. BTTB Structure. Because these are usually large-scale problems, it is useful to consider the structure of the matrix \mathbf{A} . For instance, when considering a spatially invariant blur and assuming that the image has zero boundary conditions, the matrix \mathbf{A} exhibits a Block-Toeplitz-Toeplitz-Block (BTTB) structure [21] in which \mathbf{A} appears as a $q \times q$ block-Toeplitz matrix with each block being a $p \times p$ Toeplitz matrix. That is,

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_{-1} & \mathbf{A}_{-2} & \dots & \mathbf{A}_{-(q-1)} \\ \mathbf{A}_1 & \mathbf{A}_0 & \mathbf{A}_{-1} & \dots & \mathbf{A}_{-(q-2)} \\ \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \dots & \mathbf{A}_{-(q-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{q-1} & \mathbf{A}_{q-2} & \mathbf{A}_{q-3} & \dots & \mathbf{A}_0 \end{pmatrix},$$

where for $\ell = -(q-1), -(q-2), \dots, q-1$, we have

$$\mathbf{A}_\ell = \begin{pmatrix} a_0^\ell & a_{-1}^\ell & a_{-2}^\ell & \dots & a_{-(p-1)}^\ell \\ a_1^\ell & a_0^\ell & a_{-1}^\ell & \dots & a_{-(p-2)}^\ell \\ a_2^\ell & a_1^\ell & a_0^\ell & \dots & a_{-(p-3)}^\ell \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{p-1}^\ell & a_{p-2}^\ell & a_{p-3}^\ell & \dots & a_0^\ell \end{pmatrix}.$$

Notice that to generate these matrices, we only need the first row and column of each matrix \mathbf{A}_ℓ , which form the Toeplitz vector $\mathbf{a}^\ell = (a_{-(p-1)}^\ell, \dots, a_{p-1}^\ell)$.

8.2. BCCB Structure. Another feasible structure arises when still considering a spatially invariant blur but assuming that the image has periodic boundary conditions. In this case, the matrix \mathbf{A} has a Block-Circulant-Circulant-Block (BCCB) structure, that is, a $q \times q$ block-circulant matrix with each block being a $p \times p$ circulant matrix,

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & \dots & \mathbf{A}_q \\ \mathbf{A}_q & \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{A}_{q-1} \\ \mathbf{A}_{q-1} & \mathbf{A}_q & \mathbf{A}_1 & \dots & \mathbf{A}_{q-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_2 & \mathbf{A}_3 & \mathbf{A}_4 & \dots & \mathbf{A}_1 \end{pmatrix},$$

where for $\ell = 1, \dots, q$,

$$\mathbf{A}_\ell = \begin{pmatrix} a_1^\ell & a_2^\ell & a_3^\ell & \dots & a_p^\ell \\ a_p^\ell & a_1^\ell & a_2^\ell & \dots & a_{p-1}^\ell \\ a_{p-1}^\ell & a_p^\ell & a_1^\ell & \dots & a_{p-2}^\ell \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_2^\ell & a_3^\ell & a_4^\ell & \dots & a_1^\ell \end{pmatrix}.$$

Typically, the matrix \mathbf{A} is not formed explicitly. Notice that to generate these matrices, we only need the first row of each matrix \mathbf{A}_ℓ , $\mathbf{a}^\ell = (a_1^\ell, \dots, a_p^\ell)$, so we do not need to store every entry of the matrix \mathbf{A} , only the vectors that define the matrices \mathbf{A}_ℓ . Furthermore, BCCB matrices are diagonalizable by the two-dimensional Discrete Fourier Transform (DFT) with eigenvalues given by the Fourier transform of the first column of \mathbf{A} . Therefore, the Tikhonov solution can be computed efficiently as

$$\mathbf{X}^{\text{Tik}} = \text{IDFT}(\text{conj}(\mathbf{S}) \odot \text{DFT}(\mathbf{B}) \oslash (|\mathbf{S}|^2 + \lambda \mathbf{E})),$$

where \mathbf{X}^{Tik} is the matrix representing \mathbf{x}^{Tik} (i.e., $\mathbf{x}^{\text{Tik}} = \text{vec}(\mathbf{X}^{\text{Tik}})$), the $p \times q$ matrix \mathbf{S} contains the eigenvalues of \mathbf{A} , $\text{conj}(\cdot)$ denotes the component-wise complex conjugate, $|\mathbf{S}|^2$ is the $p \times q$ matrix whose entries are the squared magnitudes of the complex entries of \mathbf{S} , \mathbf{E} is a matrix of all ones, and \odot and \oslash denote the Hadamard product (also known as Schur product) and Hadamard division, respectively, which are the element to element operations between matrices. Notice that IDFT stands for Inverse Discrete Fourier Transform. For more details, we direct the reader to [21].

8.3. Separable Blur Operator. Another special case is when the blurring operator \mathbf{A} is separable, which means that the blurring can be separated in the horizontal and vertical directions as illustrated in Equation (1), that is, the matrix \mathbf{A} can be written as

$$\mathbf{A} = \mathbf{A}_r \otimes \mathbf{A}_c.$$

If we have the SVDs $\mathbf{A}_r = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T$ and $\mathbf{A}_c = \mathbf{U}_c \mathbf{\Sigma}_c \mathbf{V}_c^T$, we basically have the SVD of \mathbf{A} by writing

$$\mathbf{A} = \mathbf{A}_r \otimes \mathbf{A}_c \quad (6)$$

$$= (\mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T) \otimes (\mathbf{U}_c \mathbf{\Sigma}_c \mathbf{V}_c^T) \quad (7)$$

$$= (\mathbf{U}_r \otimes \mathbf{U}_c)(\mathbf{\Sigma}_r \otimes \mathbf{\Sigma}_c)(\mathbf{V}_r \otimes \mathbf{V}_c)^T, \quad (8)$$

except that the elements in the diagonal matrix $\mathbf{\Sigma}_r \otimes \mathbf{\Sigma}_c$ might not be in the decreasing order, and therefore some reordering might be needed. So, the Tikhonov solution can be efficiently computed as

$$\mathbf{X}^{\text{Tik}} =$$

$$\mathbf{V}_c (\mathbf{d}_c \mathbf{d}_r^T \odot (\mathbf{U}_c^T \mathbf{B} \mathbf{U}_r) \oslash ((\mathbf{d}_c \mathbf{d}_r^T)^2 + \lambda \mathbf{E})) \mathbf{V}_r^T,$$

where \mathbf{d}_r and \mathbf{d}_c are column vectors containing the diagonal elements of $\mathbf{\Sigma}_r$ and $\mathbf{\Sigma}_c$, respectively, and $(\mathbf{d}_c \mathbf{d}_r^T)^2$ is a $p \times q$ matrix containing the squared elements of the matrix $\mathbf{d}_c \mathbf{d}_r^T$.

Notice that if \mathbf{A}_r and \mathbf{A}_c are Toeplitz matrices, then $\mathbf{A}_r \otimes \mathbf{A}_c$ has BTTB structure, and if \mathbf{A}_r and \mathbf{A}_c are circulant matrices, then $\mathbf{A}_r \otimes \mathbf{A}_c$ has BCCB structure.

9. MULTILEVEL METHODS

Because dealing with images of large size is difficult, researchers keep working on finding ways to solve these large-scale inverse problems efficiently. Another possible way to solve these problems is by means of multilevel methods. The main idea of a multilevel method is to define a sequence of systems of equations decreasing in size,

$$\mathbf{A}^{(n)} \mathbf{x}^{(n)} = \mathbf{b}^{(n)}, \quad 0 \leq n \leq L,$$

where the superscript n denotes the n -th level ($n = 0$ being the original system), and to get either an approximate solution, a correction to a current solution, or some other information at each level, where the computational cost would be smaller than solving the original system. At each level, the right-hand side is defined by

$$\mathbf{b}^{(n+1)} = \mathbf{R}^{(n)} \mathbf{b}^{(n)}$$

and the matrix by

$$\mathbf{A}^{(n+1)} = \mathbf{R}^{(n)} \mathbf{A}^{(n)} \mathbf{P}^{(n)},$$

where $\mathbf{R}^{(n)}$ is called the *restriction operator*, which maps a vector into a lower dimensional space by, for example, sampling or averaging, and $\mathbf{P}^{(n)}$ is the *interpolation operator*, which maps a vector into a higher dimensional space.

There are many ways of defining and using such a sequence of systems of equations to solve many different mathematical problems. Some references related to multilevel methods for image deblurring applications include [9, 14, 16, 26].

9.1. Wavelet-based approach. The wavelet representation started in 1910 when Haar [32] introduced the piecewise constant functions from which one can generate orthonormal basis for the solution. An extensive literature on wavelets was lead by discoveries from Strömberg, Meyer, and culminated with the celebrated Daubechies wavelets of compact support, see for instance [12, 13, 25, 34] and references therein for more details. Here, we consider the use of wavelet transforms as restriction and interpolation operators. In particular, we will work with the Haar Wavelet Transform (HWT) because, as we will see, it preserves the structures of the matrices involved.

The one-dimensional HWT is the $p \times p$ orthonormal matrix \mathbf{W} defined by

$$\mathbf{W} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & 1 & 1 & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cdots & 1 & 1 \\ 1 & -1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & 1 & -1 & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cdots & 1 & -1 \end{pmatrix} = \begin{pmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{pmatrix},$$

where \mathbf{W}_1 and \mathbf{W}_2 have dimension $p/2 \times p$. The two-dimensional HWT can be defined in terms of the one-dimensional HWT by $\mathbf{W}_{2D} = \mathbf{W} \otimes \mathbf{W} \in \mathbb{R}^{m \times m}$ with $m = p^2$. So, we define the restriction operator by $\mathbf{R} = \mathbf{W}_1 \otimes \mathbf{W}_1 \in \mathbb{R}^{(m/4) \times m}$ and the interpolation operator by $\mathbf{P} = \mathbf{R}^T$.

To motivate the use of HWT as a restriction operator, on the left of Figure 19, we can see the coarser version of \mathbf{x}^{true} defined by $\mathbf{x}^{(1)} = \mathbf{R}\mathbf{x}^{\text{true}}$. This is a 32×32 image that still conveys most of the information in the original 64×64 image with the letter H, which demonstrates that reconstructing a coarse version may be enough for some applications. This can allow us to compress images to save storage space or bandwidth. For instance, suppose that before transmitting the image \mathbf{b} , we compute its coarser version $\mathbf{b}^{(1)} = \mathbf{R}\mathbf{b}$ and transmit just $\mathbf{b}^{(1)}$ (see right of Figure 19). The image $\mathbf{b}^{(1)}$ might have enough information to recover $\mathbf{x}^{(1)}$.

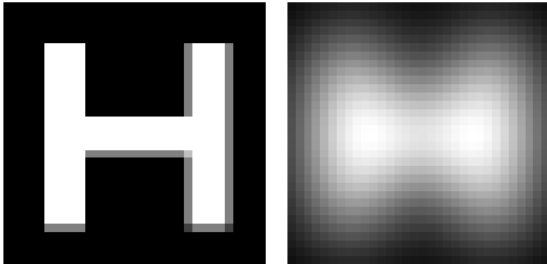


FIGURE 19. The compressed true image $\mathbf{x}^{(1)}$ on the left and blurred and noisy image $\mathbf{b}^{(1)}$ on the right.

What is the right blurring matrix to recover $\mathbf{x}^{(1)}$ from $\mathbf{b}^{(1)}$? Using our example, we will show that using $\mathbf{A}^{(1)} = \mathbf{R}\mathbf{A}\mathbf{P}$ does a good job. Figure 20 shows the Tikhonov and TV solutions of the system $\mathbf{A}^{(1)}\mathbf{x} = \mathbf{b}^{(1)}$. Notice that we are solving a system with a $1,024 \times 1,024$ matrix instead of the original system with a $4,096 \times 4,096$ matrix.

The following theorem shows that the HWT preserves the structures of the matrices described before.

Theorem 1. [16] *Let \mathbf{T} be a $p \times p$ matrix with Toeplitz structure and with Toeplitz vector \mathbf{t} , and $p = 2^s$. Then, the $2^{s-1} \times 2^{s-1}$ matrix $\mathbf{T}^1 = \mathbf{W}_1 \mathbf{T} \mathbf{W}_1^T$ is also Toeplitz with Toeplitz vector $\mathbf{t}_1 = \tilde{\mathbf{t}}(1 : 2 : 2p - 3)$, where $\tilde{\mathbf{t}} = \tilde{\mathbf{T}}\mathbf{t}$ with $\tilde{\mathbf{T}}$ being the Toeplitz matrix with Toeplitz vector with all zeros and $\mathbf{t}_0 = \mathbf{t}_{-2} = 1/2$ and $\mathbf{t}_{-1} = 1$.*

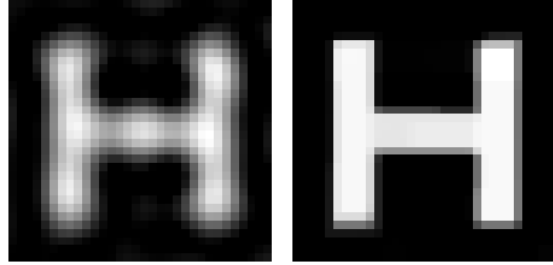


FIGURE 20. The reconstructed coarse images \mathbf{x}^{reg} of size 32×32 , obtained using the optimal regularization parameter and the discretization of the first derivative operator with the 2-norm regularization on the left, and TV regularization on the right.

A similar result can be shown for circulant matrices.

Corollary 1. *Let \mathbf{C} be a $p \times p$ circulant matrix and $p = 2^s$. Then, the $2^{s-1} \times 2^{s-1}$ matrix $\mathbf{C}_1 = \mathbf{W}_1 \mathbf{C} \mathbf{W}_1^T$ is also circulant.*

Let us consider the case when $\mathbf{A} = \mathbf{A}_r \otimes \mathbf{A}_c \in \mathbb{R}^{m \times m}$. Then, by a property of the Kronecker product, we have that

$$\begin{aligned} \mathbf{A}^{(1)} &= (\mathbf{W}_1 \otimes \mathbf{W}_1)(\mathbf{A}_r \otimes \mathbf{A}_c)(\mathbf{W}_1 \otimes \mathbf{W}_1)^T \\ &= (\mathbf{W}_1 \mathbf{A}_r \mathbf{W}_1^T) \otimes (\mathbf{W}_1 \mathbf{A}_c \mathbf{W}_1^T) \\ &= \mathbf{A}_r^{(1)} \otimes \mathbf{A}_c^{(1)}. \end{aligned}$$

The matrix $\mathbf{A}^{(1)} \in \mathbb{R}^{(m/4) \times (m/4)}$ is separable. Furthermore, by Theorem 1, if \mathbf{A}_r and \mathbf{A}_c are Toeplitz matrices, then $\mathbf{A}_r^{(1)}$ and $\mathbf{A}_c^{(1)}$ are too. Applying this same argument repeatedly, we obtain that $\mathbf{A}^{(n)}$ is separable by

two Toeplitz matrices and therefore BTTB for all levels $n = 0, \dots, L$. Similarly, by Corollary 1, $\mathbf{A}^{(n)}$ is separable by two circulant matrices and therefore BCCB for all levels $n = 0, \dots, L$. Therefore, the initial structure of the matrix is inherited by all the levels.

For the case when we have BCCB structures, we can solve the corresponding Tikhonov systems at all levels using Fourier-based methods. If we are dealing with separable matrices with Toeplitz structure, we could go down several levels until we can compute the SVD of $\mathbf{A}_r^{(n)} = \mathbf{U}_r^{(n)} \mathbf{\Sigma}_r^{(n)} (\mathbf{V}_r^{(n)})^T$ and $\mathbf{A}_c^{(n)} = \mathbf{U}_c^{(n)} \mathbf{\Sigma}_c^{(n)} (\mathbf{V}_c^{(n)})^T$, and use that

$$\begin{aligned} \mathbf{A}^{(n)} &= \mathbf{A}_r^{(n)} \otimes \mathbf{A}_c^{(n)} \\ &= (\mathbf{U}_r^{(n)} \mathbf{\Sigma}_r^{(n)} (\mathbf{V}_r^{(n)})^T) \otimes (\mathbf{U}_c^{(n)} \mathbf{\Sigma}_c^{(n)} (\mathbf{V}_c^{(n)})^T) \\ &= (\mathbf{U}_r^{(n)} \otimes \mathbf{U}_c^{(n)}) (\mathbf{\Sigma}_r^{(n)} \otimes \mathbf{\Sigma}_c^{(n)}) (\mathbf{V}_r^{(n)} \otimes \mathbf{V}_c^{(n)})^T. \end{aligned}$$

This gives us basically the SVD of $\mathbf{A}^{(n)}$, except that the elements in the diagonal matrix $\mathbf{\Sigma}_r^{(n)} \otimes \mathbf{\Sigma}_c^{(n)}$ might not be in the decreasing order, and therefore some reordering would be needed.

10. CONCLUSIONS AND OUTLOOK

Our intention has been to introduce readers to the problem of image deblurring, the mathematical issues that arise, and a few techniques for addressing them. As mentioned in the introduction, these techniques may be applied to a wide range of related problems. Indeed, we outlined a number of alternative strategies, for example, in choosing appropriate boundary conditions or in selecting the best regularization parameter, as some strategies are better suited to a specific range of applications.

Most of the methods that we have described are primarily based around direct factorizations. However, when

dealing with unstructured and very large matrices, iterative methods may need to be used instead. There is an extensive literature on iterative methods [31], but a discussion about them is outside the scope of this introductory article.

More recently, methods that are based on learning when training data are available are used to define an optimal regularization parameter, see for instance [11, 10].

We believe this subject is accessible to both undergraduate and graduate students and can serve as a good introduction to inverse problems, working with ill-conditioned linear operators, and large-scale computation. The visual nature of the problem provides compelling motivation for students and allows the efficacy of various techniques to be easily assessed. Some introductory books to this subject include [20, 21, 35].

In addition, this field is an active area of research that is connected to recent developments in machine learning and convolutional neural networks. For example, many classical techniques that have traditionally been used in image deblurring can be used to speed up the training of machine learning models, such as neural nets designed to classify images. Earlier we mentioned that Tikhonov regularization is related to ridge regression, a fundamental tool in machine learning. Indeed, the connection to machine learning goes much deeper.

The code to generate the figures appearing in this article is available at https://github.com/ipiasu/AMS_Notices_AEP. This manuscript contains only a limited number of references because journal rules restrict the maximum number of references to 20; additional references can also be found in the github link.

REFERENCES

- [1] Messier 100 shows improvements of Hubble. <https://esahubble.org/images/potw1850b/>.
- [2] Harbir Antil, Zichao Wendy Di, and Ratna Khatri. Bilevel optimization, deep learning and fractional Laplacian regularization with applications in tomography. *Inverse Problems*, 36(6):064001, 2020.
- [3] Alessandro Buccini, Mirjeta Pasha, and Lothar Reichel. Linearized Krylov subspace Bregman iteration with nonnegativity constraint. *Numerical Algorithms*, pages 1–24, 2020.
- [4] Alessandro Buccini, Mirjeta Pasha, and Lothar Reichel. Modulus-based iterative methods for constrained $\ell_p - \ell_q$ minimization. *Inverse Problems*, 36(8):084001, 2020.
- [5] Alessandro Buccini and Lothar Reichel. An $\ell^2 - \ell^q$ regularization method for large discrete ill-posed problems. *Journal of Scientific Computing*, 78:1526–1549, 2019.
- [6] Jian-Feng Cai, Stanley Osher, and Zuowei Shen. Linearized Bregman iterations for compressed sensing. *Mathematics of Computation*, 78:1515–1536, 2009.
- [7] Jian-Feng Cai, Stanley Osher, and Zuowei Shen. Linearized Bregman iterations for frame-based image deblurring. *SIAM Journal on Imaging Sciences*, 2:226–252, 2009.
- [8] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.
- [9] Raymond H Chan and Ke Chen. A multilevel algorithm for simultaneously denoising and deblurring images. *SIAM Journal on Scientific Computing*, 32(2):1043–1063, 2010.
- [10] Julianne Chung, Matthias Chung, Silvia Gazzola, and Mirjeta Pasha. Efficient learning methods for large-scale optimal inversion design. *arXiv preprint arXiv:2110.02720*, 2021.

- [11] Julianne Chung and Malena I Español. Learning regularization parameters for general-form Tikhonov. *Inverse Problems*, 33(7):074004, 2017.
- [12] Ingrid Daubechies. *Ten lectures on wavelets*. SIAM, 1992.
- [13] I Daubechies DGM, A Grossmann, and Y Meyer. Painless nonorthogonal expansions. *J. Math. Phys.*, 27:1271–1283, 1986.
- [14] Marco Donatelli and Stefano Serra-Capizzano. On the regularizing power of multigrid-type algorithms. *SIAM Journal on Scientific Computing*, 27(6):2053–2076, 2006.
- [15] Heinz Werner Engl. Discrepancy principles for Tikhonov regularization of ill-posed problems leading to optimal convergence rates. *Journal of optimization theory and applications*, 52(2):209–215, 1987.
- [16] Malena I Español and Misha E Kilmer. Multilevel approach for signal restoration problems with toeplitz matrices. *SIAM Journal on Scientific Computing*, 32(1):299–319, 2010.
- [17] Gene H Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.
- [18] Gene H Golub and Charles F Van Loan. *Matrix Computations*, 4th ed. Johns Hopkins University Press, Baltimore, 2013.
- [19] Per Christian Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. SIAM, 1998.
- [20] Per Christian Hansen. *Discrete inverse problems: insight and algorithms*. SIAM, 2010.
- [21] Per Christian Hansen, James G Nagy, and Dianne P O’leary. *Deblurring images: matrices, spectra, and filtering*. SIAM, 2006.
- [22] Per Christian Hansen and Dianne Prost O’Leary. The use of the l-curve in the regularization of discrete ill-posed problems. *SIAM journal on scientific computing*, 14(6):1487–1503, 1993.
- [23] Jie Huang, Marco Donatelli, and Raymond H Chan. Nonstationary iterated thresholding algorithms for image deblurring. *Inverse Problems & Imaging*, 7, 2013.
- [24] Marie Kubínová and James G Nagy. Robust regression for mixed Poisson–Gaussian model. *Numerical Algorithms*, 79(3):825–851, 2018.
- [25] Yves Meyer. *Wavelets and Operators: Volume 1*. Number 37. Cambridge university press, 1992.
- [26] Serena Morigi, Lothar Reichel, Fiorella Sgallari, and Andriy Shyshkov. Cascadic multiresolution methods for image deblurring. *SIAM Journal on Imaging Sciences*, 1(1):51–74, 2008.
- [27] Mirjeta Pasha, Arvind K Saiibaba, Silvia Gazzola, Malena I Espanol, and Eric de Sturler. Efficient edge-preserving methods for dynamic inverse problems. *arXiv preprint arXiv:2107.05727*, 2021.
- [28] Rosemary A Renaut, Saeed Vatanikhah, and Vahid E Ardestani. Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems. 39(2):B221–B243, 2017.
- [29] Paul Rodriguez and Brendt Wohlberg. An iteratively reweighted norm algorithm for total variation regularization. In *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, pages 892–896. IEEE, 2006.
- [30] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.
- [31] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [32] Radomir S Stanković and Bogdan J Falkowski. The Haar wavelet transform: its status and achievements. *Computers & Electrical Engineering*, 29(1):25–44, 2003.
- [33] Jean-Luc Starck and Fionn Murtagh. Astronomical image and data analysis. 2007.
- [34] Jan-Olov Strömberg and Alberto Torchinsky. Weights, sharp maximal functions and Hardy spaces. *Bulletin (New Series) of the American Mathematical Society*, 3(3):1053–1056, 1980.
- [35] Curtis R Vogel. *Computational methods for inverse problems*. SIAM, 2002.

¹ DEPARTMENT OF MATHEMATICS, GRAND VALLEY STATE UNIVERSITY, ALLENDALE, MI.

² SCHOOL OF MATHEMATICAL AND STATISTICAL SCIENCES, ARIZONA STATE UNIVERSITY, TEMPE, AZ.

³ DEPARTMENT OF MATHEMATICS, TUFTS UNIVERSITY, MEDFORD, MA.

E-mail addresses: austind@gvsu.edu, malena.espanol@asu.edu, mirjeta.pasha@tufts.edu