Reduced Noise and Ringing for Image Processing

Binan Gu[1], Burt Tilley[1], Fei Cao[2], Marina Chugunova[3], Clayton Coonrod[4], Rye Fought[5], Tyler Fuller[6], Anton Iatcenko[7], Christopher Koh[8], Trishala Thakur[9], William Thompson[4], Nikki Xu[10], and Maxim Zyskin[11]

[1]Worcester Polytechnic Institute, Department of Mathematical Sciences, Worcester, Massachusetts, USA
[2]University of Massachusetts Amherst, Department of Mathematics and Statistics, Amherst, Massachusetts, USA
[3]Claremont Graduate University, The Institute of Mathematical Sciences (IMS), Claremont, California, USA
[4]Case Western University, Department of Mathematics, Cleveland, Ohio, USA
[5]University of Vermont, Department of Mathematics and Statistics, Burlington, Vermont, USA
[6]Arizona State University, Department of Mathematics, Tempe, Arizona, USA
[7]Simon Fraser University, Department of Mathematics, Burnaby, British Columbia, Canada
[8]The University of Arizona, Department of Mathematics, Tucson, Arizona, USA
[9]University of Colorado Boulder, Department of Computer Science, Boulder, Colorado, USA
[10]North Carolina State University, Department of Mathematics, Raleigh, North Carolina, USA
[11]Oxford University, Department of Engineering Science, Oxford, United Kingdom

(*Communicated to* MIIR *on Jan 2025*)

Study Group: 40th Mathematical Problems in Industry Workshop, University of Vermont, Burlington, June 25–29, 2024

Communicated by: Taras Lakoba

Industrial Partner: The Raytheon Company, Tuscon, Arizona

Presenter: Jonathan Cain, Patrick O. Kano, Sam Yu Wang

Team Members: Binan Gu, Burt Tilley, Worcester Polytechnic Institute; Maxim Zyskin, University of Oxford; Fei Cao, University of Massachusetts Amherst; Rye Fought, William Thompson, University of Vermont; Marina Chugunova, Claremont Graduate University; Thrishala Thakur, University of Colorado Boulder; Nikki Xu, North Carolina State University; Anton Iatcenko, Simon Fraser University; Tyler Fuller, Arizona State University; Clayton Coonrod, Case Western University; Christopher Koh, The University of Arizona;

Industrial Sector: Image Processing

Key Words: Image processing; complex analysis; Gaussian deconvolution; Wiener deconvolution; regularized optimization; control theory; sparse deblurring.

MSC2020 Codes: 94A08, 68U10, 44A10, 65K10, 65E05, 49N10

Summary

The objective of this work is to devise a process that recovers an image that has been blurred by convolution with a point spread function (PSF) in a more robust manner than is currently possible with the Nearest-Neighbor Pixel Deconvolution (NNPD) algorithm proposed by Wang & Lu [11]. While PSF blurring is a fundamental limitation of optical imaging devices, it can be compensated for to some degree by utilizing *a priori* knowledge of the PSF. The NNPD algorithm at its core is a Fourier transform-based method that allows for deblurring by dividing by the transform of the PSF. The possibility of division-by-zero is mitigated in the standard NNPD algorithm by employing a threshold. Justifications for this threshold are, however, limited. Instead, in this work are various approaches from linear algebra, Fourier space smoothing, Laplace transforms, and complex analysis that show promise at improving robustness. Results are presented, and suggestions for future efforts are made.

Contents

1 Introduction

Using image processing to enhance image resolution has many commercial and defense applications. Popular image-processing algorithm approaches include inverse filters and iterative algorithms. Specific examples include Wiener deconvolution, Regularized Least Squares, Linear Least Squares, and the Nearest Neighbor Pixel Deconvolution (NNPD) algorithm.

The starting point for this project is the NNPD algorithm. It belongs to a class of inverse filter-based methods. As such, it takes the Fourier transform (FT) of an image and divides it by the FT of the point spread function (PSF). The Fourier transform of the PSF is known as the optical transfer function (OTF). The point spread function is specific to the imaging device and describes the response to a perfect point source. However, during division in Fourier space, some

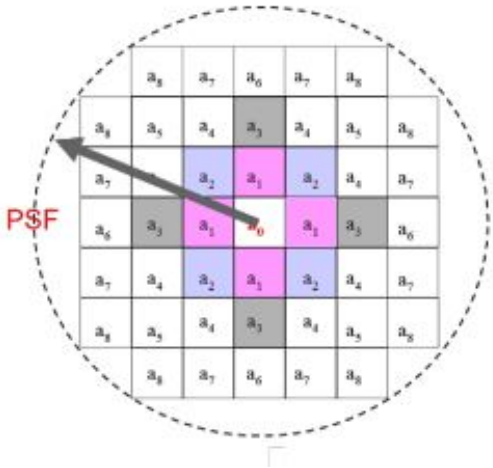


Figure 1. Nearest Neighbor Pixel Model [11].

denominators may become zero. An approach utilized in the NNPD algorithm is to introduce a threshold below which the threshold value replaces all values of the OTF. Proper threshold value selection is critical; if it is too low, it may increase noise and introduce ringing.

More precisely, a detected image I^{d} can be written as convolution of the object image I_{o} and the PSF without noise:

$$I^{\mathrm{d}} = I^{\mathrm{o}} * PSF, \tag{1.1}$$

or

$$g = f * h. \tag{1.2}$$

Taking a Fourier transform and appealing to the convolution theorem, we find that the pristine image in frequency space satisfies

$$\mathcal{F}[f] = \frac{\mathcal{F}[g]}{\mathcal{F}[h]} \implies f = \mathcal{F}^{-1}\left[\frac{\mathcal{F}[g]}{\mathcal{F}[h]}\right] \tag{1.3}$$

where the last step is typically known as the *naive deconvolution*.

One practical concern arises when $\mathcal{F}[h]$ becomes small or vanishes altogether in frequency space, creating a typical "division-by-zero" problem. To elaborate, the NNPD assumes the PSF is symmetric and uses a nearest neighbor pixel model to write the PSF according to the distance between any pixel to the PSF center as shown in Fig. 1 The PSF's center pixel has a value of a_0; its 1st order of nearest 4 neighbor pixels (left, right, up and down) have value of a_1; it's 2nd order of 4 nearest neighbors have value of a_2, and so forth. Using delta functions for each pixel, the PSF can be written as:

$$\begin{aligned} PSF_{jk} = {} & a_0\delta_{jk} + a_1\left(\delta_{j+1,k} + \delta_{j-1,k} + \delta_{j,k+1} + \delta_{j,k-1}\right) \\ & + a_2\left(\delta_{j+1,k+1} + \delta_{j-1,k-1} + \delta_{j+1,k-1} + \delta_{j-1,k+1}\right) + \cdots \end{aligned} \tag{1.4}$$

The Optical Transfer Function (OTF), now as the discrete Fourier transform of (1.4) (or simply $\mathcal{F}[h]$ in (1.3)), results in a cosine series owing to the shift theorem of Fourier transform. The

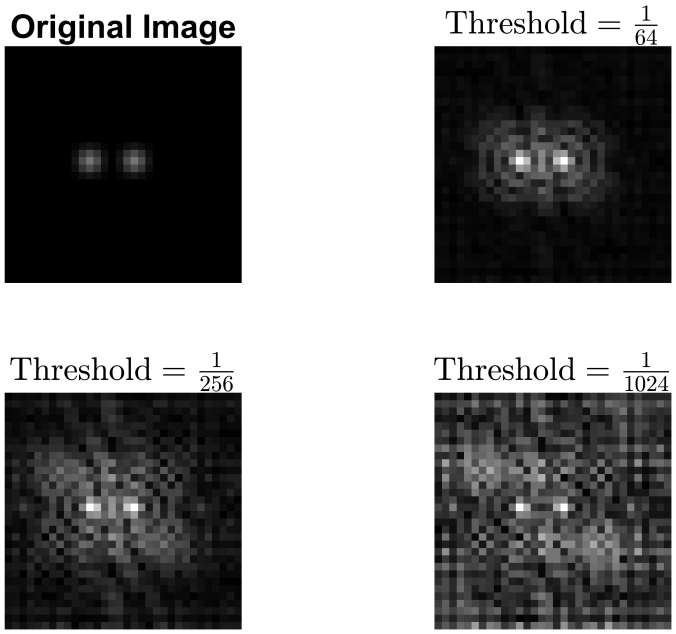


Figure 2. Reconstruction using NNPD with different threshold levels. Top left: original image.

explicit form of the OTF is given by

$$\mathcal{F}[h]_{lm} := OTF_{lm} = a_0 + 2a_1 \left[\cos\left(\frac{2\pi m}{N}\right) + \cos\left(\frac{2\pi l}{N}\right)\right] \tag{1.5}$$

$$+2a_2 \left[\cos\left(\frac{2\pi\,(l+m)}{N}\right) + \cos\left(\frac{2\pi\,(l-m)}{N}\right)\right] + \cdots . \tag{1.6}$$

Though this expression can be directly substituted back to (1.3) for deconvolution, the cosine series admits countably many zeros in the Fourier domain, contributing to the infamous division-by-zero issue. Wang & Lu [11] used a lower threshold to bound the OTF away from zero as a remedy. While the resulting deconvolution has enhanced the sharpness of the target (see figure 2), it has also produced noticeable noise and ringing.

Analytical approaches to (1.3) center on deforming the contour of integration for the inverse Fourier transform to avoid division by zero. However, converting an actual image into Fourier space results in the Fourier transform over a discrete mesh. Deforming the contour results in finding interpolated values of the Fourier transform as part of the calculation of the inverse, which has the potential to increase the noise to the original image (see Greengard & Lee [4] on techniques to resolve nonuniform Fourier transforms). This class of problems is known to be ill-posed, so the approaches for the solution require care.

Below we describe various techniques to perform (1.3) in a mathematically robust way. Section 2 focuses on using Laplace transforms in place of the Fourier transform via Weeks' Method, the Fixed Talbot's Method, and the Bi-Lateral Laplace Transform. Section 3 attempts to use Cauchy integrals to improve signal clarity. Section 4 centers on different regularization techniques to prevent the division by zero condition. Section 5 investigates modifications of the continuous analogue of the NNPD. Section 6 looks at a deblurring technique using dictionary learning. We conclude in Section 7.

2 Laplace transform method

One approach to circumvent the numerical instability inherent to Fourier deconvolution in the NNPD is utilizing the more general Laplace transform. The Fourier transform can be viewed as a special case of the Laplace transform. Consideration, however, must be given to the sense of the transformation. While applying the Fourier transform to spatial problems is standard, Laplace transforms are typically utilized for causal time-dependent problems. For example, while both transforms obey the convolution-to-product property, the casual Laplace transform convolution integral has bounds from 0 to time t. Two approaches have been considered to address this fundamental difference.

One is to utilize numerical inversion methods with the standard convolution sense and consider the radial aspect of the transform. That is to utilize polar coordinates in the image and use Fourier transforms in θ but Laplace in radial r. Existing numerical inversion methods, such as Weeks' and Talbot's method, have been employed.

A second approach is to consider the bilateral Laplace transform. The bilaterial Laplace transform also obeys the convolution-to-product property and is not casual.

$$\mathrm{L}_b(s) = \int_{-\infty}^{\infty} e^{-st} f(t) dt \tag{2.1}$$

The inverse bilateral Laplace transform is also defined by the Bromwich contour Γ.

$$f(t) = \frac{1}{2\pi i} \int_{\Gamma} e^{st} F(s) dt \tag{2.2}$$

However, unlike the standard standard Laplace transform which has the half plane to the left of the contour as the region of convergence, the region of convergence for the bilateral Laplace transform is typically much more restrictive. This reflects the fact that the integral much converge for both the positive and negative time t.

2.1 Weeks' method

Contributed by Anton Iatcenko and Marina Chugunova

Due to time constraints, we only performed a one-dimensional version of the deblurring procedure. Beginning with a sharp image $u_0(t)$ and a given blurring kernel $\kappa(t)$ (a decaying exponential), we construct a synthetic blurred image by a causal convolution:

$$u_{\mathrm{blrd}}(t) = \int_0^t u_0(t)\kappa(t-\tau)\, d\tau = \int_0^t u_0(t) e^{-a(t-\tau)}\, d\tau. \tag{2.3}$$

The (one-dimensional) image is treated as a piece-wise constant function, so its Laplace transform can be computed exactly:

$$\hat{u}_{\mathrm{blrd}}(s) = \frac{1}{s} \sum_{j=1}^{N} u_j \left[e^{-t_{j-1}s} - e^{-t_j s} \right]. \tag{2.4}$$

The Laplace transform of the smoothing kernel is also computed exactly, from which the deblurred image in Laplace space is computed by setting

$$\hat{u}_{\text{de-blrd}}(s) = \frac{\hat{u}_{\mathrm{blrd}}(s)}{\hat{\kappa}(s)}, \tag{2.5}$$

from which the image in real space is obtained by inverting the Laplace transform using Weeks'

method [5]. It assumes the signal is represented in the Laguerre basis:

$$u_{\text{de-blrd}}(t) = e^{\sigma t} \sum a_n e^{-bt} L_n(2bt), \tag{2.6}$$

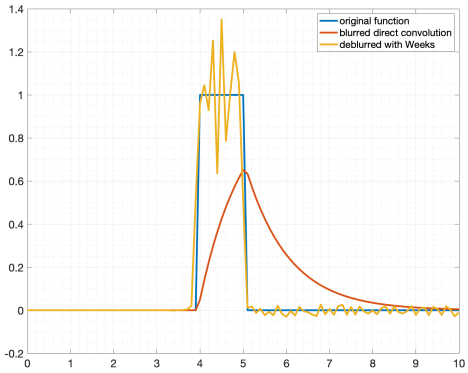
where the coefficients are computed using the Bromwich contour, which results in the formula

$$a_n = \frac{e^{-n\pi/(2M)}}{2M} \sum_{m=M}^{M-1} \left(\frac{2be^{-in\theta_m}}{1 - e^{i\theta_{m+1/2}}} \right) \hat{u}_{\text{de-blrd}} \left(\sigma - b \frac{e^{i\theta_{m+1/2}} + 1}{e^{i\theta_{m+1/2}} - 1} \right),$$

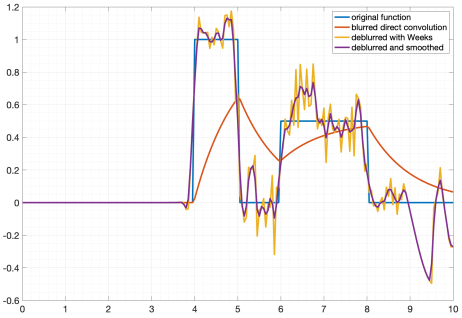
where σ and b are free parameters chosen to minimize the estimated error.

Some results are presented in Figure 3 below: we see that the results of de-blurring exhibit some sharp oscillations not present in the original image. This can likely be attributed to a Gibbs phenomenon: as the original discontinuous function is represented by a sum of Laguerre polynomials, some oscillations near the edges are expected.

We also note that the effective supports of the pulses are recovered fairly well in both cases. In particular, in test case 3b, we do recover a good separation between two distinct parts of the image, which was lost almost entirely to blurring.



(a) De-blurring a single pulse



(b) De-blurring a double pulse

Figure 3. Pictures of animals

2.2 Fixed Talbot's method

Contributed by Anton Iatcenko and Marina Chugunova

From the Matlab code for the classical Talbot's method [5], which was based on high precision computations and used 1000 digits to get accuracy for inverse Laplace transform of Gaussian at the level 10^{-17}. Researching modern articles related to modifications and improvements to Talbot's method, we found that Fixed Talbot's method algorithm was used. We modified the Matlab code, and using only 25 digits could get an accuracy level of 10^{-15}. The computational time for the original code was as long as 270 seconds, and the modified one was down to just 35 seconds.

In 1979, Talbot pioneered the approach to numerical Laplace inversion by deforming the standard contour in the Bromwich integral [9]. Only a few such algorithms have been developed along these lines [7, 1, 2]. It seems surprising how little attention this approach has received in the literature relative to other methods. Unlike the Fourier series or Post-Widder approaches,

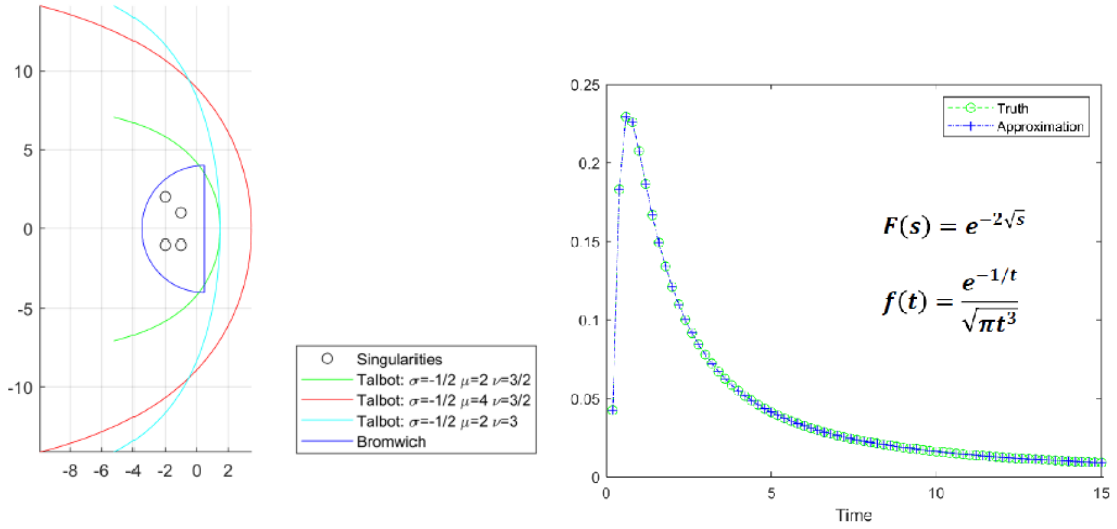
Talbot's method idea is to replace the contour with one that opens toward the negative real axis:

$$s(\theta) = \sigma + \lambda\theta(iv + \cot(\theta))$$

where $\lambda, v, \sigma \in \mathbb{R}$ and $\theta \in (-\pi, \pi)$.

It also requires that:

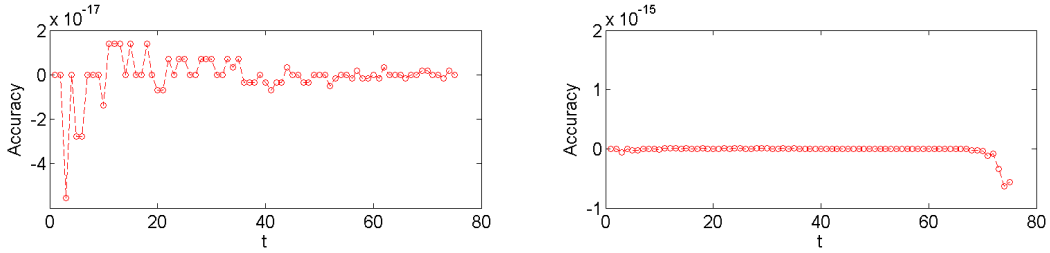
- $|F(s)| \to 0$ uniformly as $|s| \to \infty$
- $\Re(s_j) < K$ for all singularities s_j



(a) Talbot's contours

(b) Relative accuracy plot in %

Figure 4. Illustration of the Talbot's Method



(a) Talbot method

(b) Fixed Talbot method

Figure 5. On the left: Talbot method: high precision computations with 1000 digits, computational time 270 seconds. On the right: Fixed Talbot method: high precision computations with 25 digits, computational time 35 seconds.

It thus inherently assumes that the physical system damps highly oscillatory terms and is not appropriate for purely conservative problems. From a numerical perspective, however, damping the highly oscillatory terms in the inversion integral is a stabilizing procedure since it is these that provide the largest contribution to the error when one performs the numerical quadrature of the Laplace inversion integral.

The primary difficulty of the Talbot method is the selection of accurate values for the contour parameters. The original paper by Talbot is rather lengthy, with only a few pages on the

derivation of contours using the steepest-descent method. The remaining pages are a detailed discussion of how to determine the parameters from the properties of the function $F(s)$ to invert. The nontrivial task of developing software to select optimal values for these parameters has been undertaken for scalar functions [7], but the determination of these coefficients relies heavily on the known location of the singularities of the Laplace space functions. In response, many attempts have been made to simplify Talbot's method or provide alternative integration contours [2, 10].

The Fixed Talbot Method, introduced by Abate and Valkó (Abate, J., Valkó, P. P. (2004). Multi-precision Laplace transform inversion. Numerical Methods for Engineering), is a simplified and efficient technique for the numerical inversion of Laplace transforms. This method builds upon the foundational work by Talbot, who proposed the idea of deforming the Bromwich contour in the complex plane to achieve more accurate numerical inversions. The Fixed Talbot Method fixes certain parameters, which makes it easier to implement and apply while still retaining high accuracy for a wide range of functions.

2.2.1 *Mathematical Formulation*

The central idea of the Fixed Talbot Method is to use a predefined contour in the complex plane for the inversion integral:

$$f(t) = \frac{1}{2\pi i} \int_B \exp(ts)\hat{f}(s)\, ds.$$

In the Bromwich integral, the contour B is a vertical line defined by $s = r + iy$, where $-\infty < y < +\infty$ and r is a fixed value chosen such that all the singularities of the transform are to the left of it. Talbot's approach involves deforming this contour to a new path in the complex plane, which can be parameterized as:

$$s(\theta) = r\theta(\cot\theta + i), \quad -\pi < \theta < \pi,$$

where r is a parameter that needs to be fixed. This parameter is crucial for balancing numerical stability and accuracy.

2.2.2 *Parameter Selection*

The selection of the parameter r is essential for the method's performance. In the Fixed Talbot Method, r is set to:

$$r = \frac{2M}{5t},$$

where M is the number of terms in the summation series and t is the time variable. This choice of r ensures that the contour captures the essential features of the integrand, thus improving the numerical inversion's accuracy and efficiency.

2.2.3 *Numerical Implementation*

The numerical inversion process using the Fixed Talbot Method involves several steps:

1. **Discretize the Contour**: The contour $s(\theta)$ is discretized into N points $\{z_k\}$ along the path defined by $s(\theta)$. 2. **Evaluate the Integrand**: The integrand is evaluated at each discretized point z_k:

$$F(z_k) = \exp(ts(\theta))\hat{f}(s(\theta))s'(\theta),$$

where $s'(\theta) = ir(1 + i\sigma(\theta))$ and

$$\sigma(\theta) = \theta + (\theta \cot \theta - 1) \cot \theta.$$

3. **Numerical Integration**: The integral is approximated using the trapezoidal rule:

$$f(t, M) = \frac{r}{M} \left[\frac{1}{2} \hat{f}(r) \exp(rt) + \sum_{k=1}^{M-1} \Re \left[\exp(ts(\theta_k)) \hat{f}(s(\theta_k))(1 + i\sigma(\theta_k)) \right] \right].$$

By fixing the parameter r, the Fixed Talbot Method simplifies the implementation, reducing the complexity of dynamically adjusting the contour for different functions.

2.2.4 *Advantages and Limitations*

The Fixed Talbot Method offers several advantages: - **Simplicity**: The method is straightforward to implement because it fixes the contour parameters. - **Efficiency**: The fixed contour allows for efficient computation, as the parameters do not need to be adjusted dynamically for different transforms. - **Accuracy**: Despite the fixed contour, the method produces accurate results for a wide range of Laplace transforms.

However, there are also limitations: - **Flexibility**: The fixed contour may not be optimal for all functions, particularly those with complex singularities or branch cuts. - **Precision**: While the method provides good accuracy for many practical purposes, it may not achieve the same level of precision as methods that adapt the contour based on the specific function being inverted.

2.2.5 *Practical Implementation*

In practical applications, the Fixed Talbot Method has been shown to be highly effective for numerical Laplace transform inversion. The method's performance has been evaluated for a large class of transforms, with the relative error estimate given by:

$$\left| \frac{f(t) - f(t, M)}{f(t)} \right| \approx 10^{-0.6M}.$$

This indicates that the number of significant digits in the approximant $f(t, M)$ is about equal to $0.6M$.

Despite its simplicity, the Fixed Talbot Method requires careful attention to potential computing problems. For instance, issues may arise with the computation of certain transforms, such as those involving branch cuts for the square root function. These problems can be mitigated by expressing the transform in a way that avoids problematic branch cuts.

2.2.6 *Conclusion*

The Fixed Talbot Method provides a practical and efficient approach for the numerical inversion of Laplace transforms. Fixing key parameters simplifies the implementation and reduces computational complexity while maintaining a high degree of accuracy. This makes it a valuable tool for engineers and scientists who require reliable numerical inversion of Laplace transforms in their work. The method's fixed contour and parameter choices ensure robust performance across a wide range of functions, demonstrating its versatility and utility in various applications.

2.3 Bi-lateral Laplace Transform

Contributed by Nikki Xu, Rye Fought, and Trishala Thakur

For a suitable function $f(t)$, the bilateral Laplace transform is defined as the double-sided improper integral

$$\mathcal{L}_b\{f\}(s) = \int_{-\infty}^{\infty} e^{-st} f(t) dt \tag{2.7}$$

and its region of convergence is often a strip in the complex domain where the real part of s is bounded, i.e., $\alpha < \Re(s) < \beta$. In particular, when this strip contains the imaginary axis, we recover the Fourier transform with the following relation

$$\mathcal{L}_b\{f\}(s = i\omega) = \mathcal{F}\{f\}(s)$$

where $\mathcal{F}$ denotes the Fourier transform.

The inverse bilateral Laplace transform is also defined by a contour integration over the contour Γ.

$$f(t) = \frac{1}{2\pi i} \int_{\Gamma} e^{st} F(s) dt \tag{2.8}$$

If choosing the Bromwich contour $\Gamma = \sigma + i\omega$, then one must distinguish between different regions of the complex plane based on the value of σ. The choice of σ can yield a different time domain inverse function. For the standard unilateral Laplace transform, the Bromwich contour is so that σ is greater than all poles, that is to the right of all poles in the complex plane. In the bilateral Laplace transform, this condition is not placed on Γ.

3 Full Signal Recovery with Cauchy integrals

Contributed by Maxim Zyskin

Let's assume blurring by scaled translates of a kernel χ,

$$I_d(\mathbf{x}) = h * I^0(\mathbf{x}) := \int \left(\sum_{\mathbf{m}} a_{\mathbf{m}} \chi(\mathbf{x} - \mathbf{x}' - \mathbf{m}) \right) I_0(\mathbf{x}') d\mathbf{x}', \tag{3.1}$$

where, in the 2d version, $\mathbf{m} = (m_1, m_2), m_i = 0, \pm 1, \pm 2, \ldots$. Fourier transform $\widehat{f}(\mathbf{k})$ of a function $f(\mathbf{x})$ on the real space is

$$f(\mathbf{x}) \to \widehat{f}(\mathbf{k}) := \int f(\mathbf{x}) e^{i\mathbf{kx}} d\mathbf{x} \tag{3.2}$$

As the Fourier transform of a convolution is a product of the Fourier transforms, and as translates by some $\mathbf{a}$ in the real space correspond to multiplications by $e^{i\mathbf{ka}}$ in the Fourier space, it follows that the Fourier transform version of (3.1) is

$$\widehat{I}_d(\mathbf{k}) = \widehat{\chi}(\mathbf{k}) \left(\textstyle\sum_{\mathbf{m}} a_m e^{i\mathbf{km}} \right) \widehat{I}_0(\mathbf{k}), \tag{3.3}$$

(to see that, it may help to think of splitting $\mathbf{kx}$ as $\mathbf{k}(\mathbf{x} - \mathbf{x}' + \mathbf{m}) + \mathbf{kx}' + \mathbf{km}$ while Fourier transforming summands in (3.1)) Thus at least formally, the reconstructed image I_0 in the Fourier space is given by

$$\widehat{I}_0(\mathbf{k}) = \frac{\widehat{I}_d(\mathbf{k})}{\widehat{\chi}(\mathbf{k}) \left(\sum_{\mathbf{m}} a_m e^{i\mathbf{km}} \right)} \tag{3.4}$$

and in the real space is a convolution of the blurred image with the kernel h_{inv}, whose Fourier symbol is

$$\widehat{h}_{inv} = \frac{1}{\widehat{\chi(\mathbf{k})}} \frac{1}{\sum_{\mathbf{m}} a_m e^{i\mathbf{km}}} \tag{3.5}$$

Here, we assume that the true image is sufficiently nice and that the standard Fourier inversion formula is applicable. But for the kernel h_{inv} itself, as $\widehat{h}_{inv}$ may have singularities for real values of $\mathbf{k}$ or not have sufficiently fast decay as $\mathbf{k} \to \infty$, the inverse transform shall be understood in the sense of generalized functions, on a sufficiently nice space of test functions.

As Fourier products correspond to convolutions in the real space, we can treat one at a time the shape factor $\frac{1}{\widehat{\chi(\mathbf{k})}}$ and the translation factor $\frac{1}{\sum_{\mathbf{m}} a_m e^{i\mathbf{km}}}$ in the kernel $\widehat{h_{inv}}$.

We discuss below the inverse Fourier transform of the translation factor $\frac{1}{\sum_{\mathbf{m}} a_m e^{i\mathbf{km}}}$ (this corresponds to the shape function χ being a delta-function, and its Fourier transform = 1). Note that

$$e^{i\mathbf{km}} = {z_1}^{m_1} {z_2}^{m_2}, z_1 := e^{ik_1}, z_2 := e^{ik_2}, \tag{3.6}$$

Thus, we would want to make sense of

$$\begin{aligned} h_{1,inv} &:= \int \frac{e^{-i\mathbf{kx}}}{W(\mathbf{k})} \frac{d\mathbf{k}}{(2\pi)^d}, \\ W(\mathbf{k}) &:= \textstyle\sum_{\mathbf{m}} a_m e^{i\mathbf{km}} = \sum_{\mathbf{m}} a_m \mathbf{z}^{\mathbf{m}}, \quad \mathbf{z}^{\mathbf{m}} := {z_1}^{m_1} {z_2}^{m_2}. \end{aligned} \tag{3.7}$$

Thus zeroes of $W = \sum a_{\mathbf{m}} {z_1}^{m_1} {z_2}^{m_2}$ are determined, in the 2d case, by zeroes of a polynomial in z_1, z_2 variables. One of the integrations in (3.7) can be performed using residues at zeroes of W (while keeping the second integration variable fixed). That would typically produce algebraic factors having branch cuts in the remaining integration variable, for example (3.11), where D is a polynomial in the remaining integration variable, thus the second integration needs other techniques or a numerical evaluation. (The convergence of the latter would be improved if in the remaining integration there are no singularities near the integration contour).

Let us now consider the simplified case of 1-dimensional integration and further assume that only $m = 0, \pm 1$ terms are present:

$$W = a_0 + a_1 z + a_{-1} 1/z = \frac{a_1}{z} (z - z_+) (z - z_-), z_{+,-} := \frac{-a_0 \pm \sqrt{a_0^2 - 4a_1 a_{-1}}}{2a_1} \tag{3.8}$$

(note that in the 2d case those a_i are polynomials in z_2, $1/z_2$).

Further, assume that the discriminant $D = a_0^2 - 4a_1 a_{-1}$ is positive (or negative but $a_1 \neq a_{-1}$); in that case, there are no zeroes of W corresponding to k on the real axes, or equivalently, with z on the unit circle. Note that away from zeroes of W, in particular on the large k-semicircles in the complex plane, $1/W$ is bounded, and e^{-ikx} is small and $\to 0$ as $|k| \to \infty$, on the upper (lower) k-semicircle when, respectively, $x > 0$ ($x < 0$); thus we may close the k contour by a semi-circle in the corresponding half-plane, and compute the k integral by residues. Assume $D > 0$ and $a_0 > 0$. Then the roots $z_+, z_- < 0$, and corresponding $k_\pm$ are given by

$$k_\pm = \{-i \; log \, |z_\pm| + \pi(2n+1), n = 0, \pm 1, \pm 2, \ldots\}. \tag{3.9}$$

Further, assume that $a_\pm > 0$, and a_0 is sufficiently larger than $a_\pm$, so that in particular $a_0 > a_1 + a_{-1}$. In that case, $z_- < -1, -1 < z_+ < 0$, and $Im \; k_+ > 0, Im \; k_- < 0$ in (3.9).

The residues (with respect to k) at z_+ are $\frac{e^{-ik_+ x}}{ia_1(z_+ - z_-)}$; this can be equivalently expressed as:

$$res_{k_\pm} = \pm \frac{e^{-ik_\pm x}}{i\sqrt{D}}, D = a_0^2 - 4a_1 a_{-1}. \tag{3.10}$$

Closing the integration contour by large circles in the corresponding half k-planes, the Cauchy theorem implies that

$$h_{1,inv}(x) = \frac{e^{i\pi x}}{|z_\mp|^x \sqrt{D}} \sum_n e^{-2\pi i n x} = \frac{(-1)^x}{|z_\mp|^x \sqrt{D}} \sum_m \delta(x+m), x \gtrless 0 \tag{3.11}$$

Let's see how this works:
Image: $f(x)$, assumed to be continuous and decaying sufficiently fast as $|x| \to \infty$.
Blurred image:

$$d(x) = a_0 f(x) + a_1 f(x-1) + a_{-1} f(x+1). \tag{3.12}$$

Recovering:

$$\begin{aligned}
&\sqrt{D}\; h_{1,inv} * d(x) = \\
&= d(x) + \textstyle\sum_{m=1}^{\infty} \frac{(-1)^m d(x-m)}{|z_-|^m} + \sum_{m=1}^{\infty} (-1)^m d(x+m)|z_+|^m = \\
&= a_0 \left(f(x) + \textstyle\sum_{m=1}^{\infty} \frac{(-1)^m f(x-m)}{|z_-|^m} + \sum_{m=1}^{\infty} (-1)^m f(x+m)|z_+|^m \right) + \\
&a_1 \left(f(x-1) - |z_-| \textstyle\sum_{m=2}^{\infty} \frac{(-1)^m f(x-m)}{|z_-|^m} - |z_+|\, f(x) - |z_+| \sum_{m=2}^{\infty} (-1)^m f(x+m)|z_+|^m \right) + \\
&a_{-1} \left(f(x+1) - \frac{f(x)}{|z_-|} - \frac{1}{|z_-|} \textstyle\sum_{m=2}^{\infty} \frac{(-1)^m f(x-m)}{|z_-|^m} - \frac{1}{|z_+|} \sum_{m=2}^{\infty} (-1)^m f(x+m)|z_+|^m \right) = \\
&= \left(a_0 - a_1 |z_+| - a_{-1} \tfrac{1}{|z_-|} \right) f(x) + \left(a_0 - a_1 |z_-| - a_{-1} \tfrac{1}{|z_-|} \right) \textstyle\sum_{m=1}^{\infty} \frac{(-1)^m f(x-m)}{|z_-|^m} + \\
&+ \left(a_0 - a_1 |z_+| - a_{-1} \tfrac{1}{|z_+|} \right) \textstyle\sum_{m=1}^{\infty} (-1)^m f(x+m)|z_+|^m,
\end{aligned} \tag{3.13}$$

where the second equality follows from (3.11); substituting $d(x)$ from (3.12) and re-labeling summation indexes accordingly, we get the next equality; combining terms and rearranging, we get the last equality. Remember that the above formulas were derived under assumptions on parameters where z_-, z_+ are both real and negative roots of W. Thus

$$\begin{aligned}
&a_0 - a_1 |z_-| - a_{-1} \tfrac{1}{|z_-|} = a_0 + a_1 z_+ + a_{-1} \tfrac{1}{z_-} = 0; \\
&a_0 - a_1 |z_-| - a_{-1} \tfrac{1}{|z_+|} = a_0 + a_1 z_+ + a_{-1} \tfrac{1}{z_+} = 0; \\
&a_0 - a_1 |z_+| - a_{-1} \tfrac{1}{|z_-|} = a_0 + a_1 z_+ + a_{-1} \tfrac{1}{z_-} = \\
&= a_0 + a_1 z_+ - (a_0 + a_1 z_-) = a_1 (z_+ - z_-) = \sqrt{D}.
\end{aligned} \tag{3.14}$$

Combining (3.13), (3.14) we conclude that hereby **the image is completely recovered !**

$$\boxed{h_{1,inv} * d(x) = f(x).} \tag{3.15}$$

Examples. Let $f(x)$ be functions with graphs given respectively, by Figures 6 (a) and 7 (a), illustrating, respectively, continuous and discontinuous with the one-sided limits cases. Let $d(x)$ be given by (3.12), with $a_0 = 0.6, a_1 = 0.15, a_{-1} = 0.25$. Plots of $d(x)$ are given, respectively, by Figures 6 (b), 7 (b). Let

$$f_r(x; M) := d(x) + \textstyle\sum_{m=1}^{M} (-1)^m \left(\frac{d(x-m)}{|z_-|^m} + d(x+m)\, |z_+|^m \right) \tag{3.16}$$

be an approximate reconstruction, with an infinite sum replaced by a sum of the first M terms. Plots of the approximate reconstruction $f_r(x; M)$ with $M = 1, 2, 4, 8$ are given, respectively, by Figures 6 (c-f), 7 (c-f). Note that in both examples, the true image was completely restored at $M = 8$.

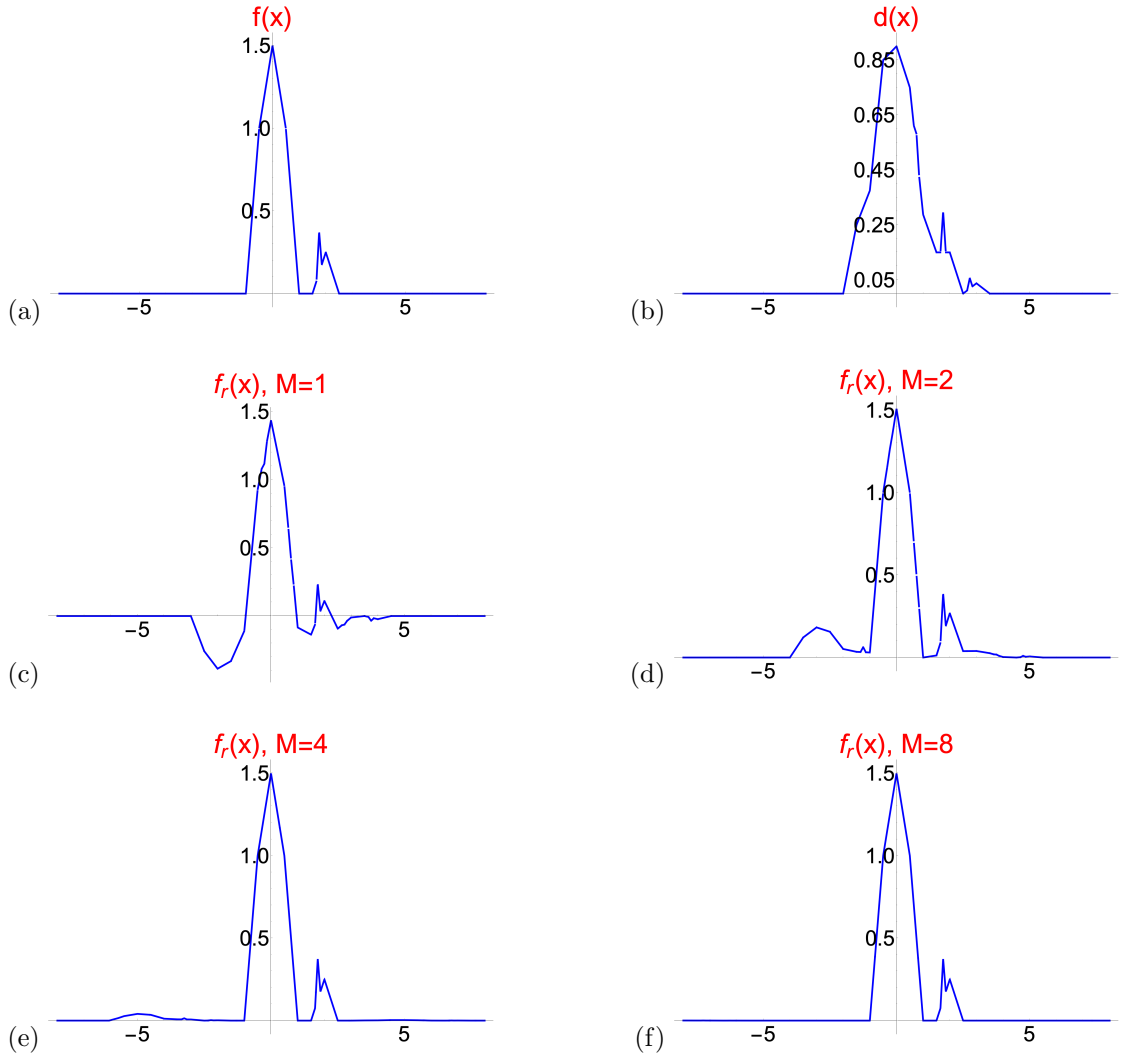
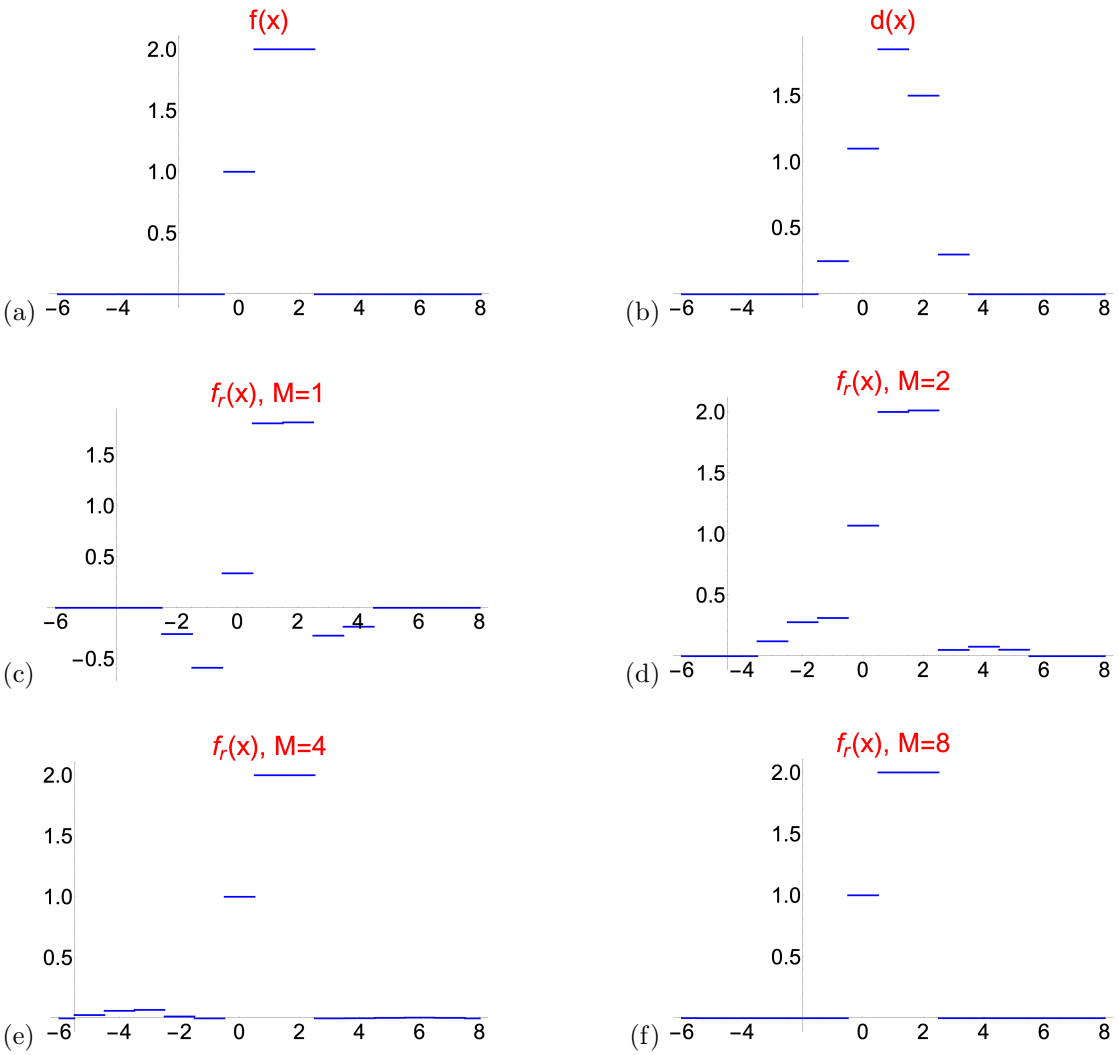


Figure 6. (a) True image $f(x)$, (b) blurred image $d(x)$, and (c-f) M terms approximations to the deblurred image $f_r(x, M), M = 1, 2, 4, 8$



(a) (b) (c) (d) (e) (f)

Figure 7. (a) True image $f(x)$, (b) blurred image $d(x)$, and (c-f) approximations to the deblurred image $f_r(x, M), M = 1, 2, 4, 8$

4 Regularized optimization technique

Let's perform a short recap of the main problem investigated in this report. Given a linear functional H and a known function $g(x, y)$ (representing the degraded/detected image), the goal is to recover the unknown function $f(x, y)$ (modeling the original image) from the following functional equation:

$$g(x, y) = \mathrm{H}[f](x, y) + n \tag{4.1}$$

where n is an additive noise, typically independent of the signal f.

This section is devoted to the designation of several regularization techniques in order to circumvent the aforementioned "division by zero" problem. Sections 4.1 and 4.2 study the original problem without noise ($n = 0$), and section 4.3 addresses the noisy case ($n \neq 0$). It is worth emphasizing that the regularization methods carried out in this section will not regularize the PSF h directly, and we defer the discussions of subsequent smoothing approaches (acting directly on the PSF or its Fourier transform) to section 5.

4.1 Tikhonov regularization

Contributed by Fei Cao and Binan Gu

We note that identity (4.1) with $n = 0$ can be rewritten as

$$g(x, y) = \iint_{\mathbb{R}^2} f(x_0, y_0)\, \mathrm{H}[\delta(x - x_0, y - y_0)]\, \mathrm{d}x_0\, \mathrm{d}y_0, \tag{4.2}$$

where the function $\mathrm{H}[\delta(x - x_0, y - y_0)] \coloneqq \mathrm{H}[\delta(\cdot - x_0, \cdot - y_0)](x, y)$ is termed as the point spread function (PSF) in image processing literature. Since H is a linear functional, we can identify it (via the Riesz representation theorem) with some h (in the dual space that contains H), meaning that for each pair $(x, y) \in \mathbb{R}^2$ there exists a unique h for which

$$\mathrm{H}[f](x, y) = \iint_{\mathbb{R}^2} f(x_0, y_0)\, h(x, y; x_0, y_0)\, \mathrm{d}x_0\, \mathrm{d}y_0.$$

Thus we can reformulate the identity (4.2) as

$$g(x, y) = \iint_{\mathbb{R}^2} f(x_0, y_0)\, h(x, y; x_0, y_0)\, \mathrm{d}x_0\, \mathrm{d}y_0. \tag{4.3}$$

In practice, the parametric function $h(x, y; x_0, y_0)$ takes the form of a kernel which we write as $h(x, y; x_0, y_0) = h(x - x_0, y - y_0)$. Consequently, we arrive at the following simplified equation

$$g(x, y) = [f * h](x, y) \tag{4.4}$$

from which one can deduce the Fourier transform of the original image:

$$\mathcal{F}[f](\xi, \eta) = \frac{\mathcal{F}[g](\xi, \eta)}{\mathcal{F}[h](\xi, \eta)}. \tag{4.5}$$

The major concern lies in the possibility of having a vanishing $\mathcal{F}[h](\xi, \eta)$ (Fourier transform of the PSF, also known as the optical transfer function (OTF)) in the Fourier domain $(\xi, \eta) \in \mathbb{R}^2$, which renders us a notorious "division by zero" issue that must be addressed carefully for practical applications.

Tikhonov regularization is a popular method originally designed to solve certain (possibly) ill-posed inverse problems. We introduce a (typically small) hyper-parameter $\varepsilon > 0$ and try to solve the following (regularized) minimization problem:

$$f_\varepsilon = \operatorname*{argmin}_{f \in L^2} \left\{ \|g - \mathrm{H}[f]\|_{L^2}^2 + \varepsilon \|f\|_{L^2}^2 \right\} \coloneqq \operatorname{argmin} \mathcal{L}_\varepsilon[f]. \tag{4.6}$$

ε	Benchmark	10^{-2}	10^{-3}	10^{-4}
SNR	82.4	21.2	22.1	19.2
Sharpness	53	79	104.5	116.6

Table 1. Performance of the Tikhonov regularized image recovery tested against the benchmark statistics, using various values for the hyper-parameter ε.

We emphasize that the choice of $\varepsilon = 0$ leads to the same inverse problem as proposed in the reference notes (whose closed-form solution in the Fourier domain (4.5) suffers a troubling "division by zero" issue). By a standard perturbation argument requiring that

$$\mathcal{L}_\varepsilon[f + v] \leq \mathcal{L}_\varepsilon[f]$$

for any negligible perturbation v, we deduce that a minimizer f_ε of $\mathcal{L}_\varepsilon[\cdot]$ must satisfy the following normal equation

$$\mathrm{H}^*[g - \mathrm{H}[f_\varepsilon]] = \varepsilon\, f_\varepsilon \tag{4.7}$$

with H^* the adjoint of H. We now recall that the linear functional H is identified via Riesz representation with some h in the previous section (i.e., $\mathrm{H}[f] = f * h$). Assuming further that the kernel h is even in both variables, i.e.,

$$h(-x, -y) = h(x, y) \tag{4.8}$$

for all $(x, y) \in \mathbb{R}^2$, then $\mathrm{H}^* = \mathrm{H}$ (meaning that the linear operator H is self-adjoint) and the equation (4.7) simplifies to

$$\mathrm{H}[g] - \mathrm{H}^2[f_\varepsilon]] = \varepsilon\, f_\varepsilon \tag{4.9}$$

or equivalently

$$g * h - (f_\varepsilon * h) * h = \varepsilon\, f_\varepsilon. \tag{4.10}$$

Lastly, we take the Fourier transformed version of (4.10) to conclude that

$$\mathcal{F}[f_\varepsilon](\xi, \eta) = \frac{\mathcal{F}[g](\xi, \eta)\, \mathcal{F}[h](\xi, \eta)}{\varepsilon + |\mathcal{F}[h](\xi, \eta)|^2}. \tag{4.11}$$

In general, if we drop the symmetric assumption (4.8) on h then the resulting expression (4.11) for $\mathcal{F}[f_\varepsilon]$ becomes

$$\mathcal{F}[f_\varepsilon](\xi, \eta) = \frac{\mathcal{F}[g](\xi, \eta)\, \overline{\mathcal{F}[h](\xi, \eta)}}{\varepsilon + |\mathcal{F}[h](\xi, \eta)|^2}. \tag{4.12}$$

where $\overline{a}$ denotes the complex conjugate of $a \in \mathbb{C}$.

We emphasize here that the expression for $\mathcal{F}[f_\varepsilon](\xi, \eta)$ avoids the "division by zero" due to the presence of the hyper-parameter $\varepsilon > 0$ in the denominator of the right-hand side of (4.11), and it boils down to our previous expression (4.5) when we send $\varepsilon \to 0$.

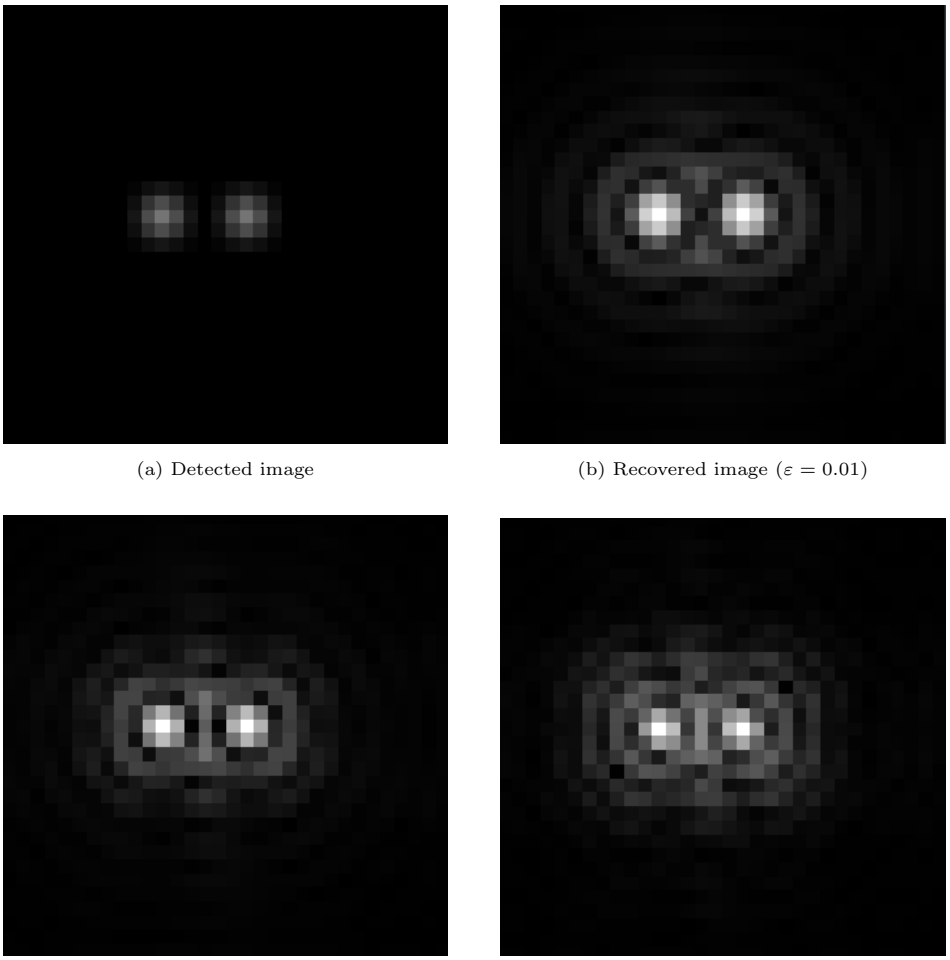
4.2 Control approach

Contributed by Pavel Dubovski

Modified problem: given $f_{\text{obs}}(x)$ find $f(x)$ given that

$$H[f] = f_{\text{obs}} + v \tag{4.13}$$

with operator $H[f](x) = f * h(x) = \int_{\mathbb{R}} f(x - y)h(y)dy$.
Changes in control v generate the change in solution f.
We consider the inverse/control problem: find v and f which satisfy (4.13).



(a) Detected image

(b) Recovered image ($\varepsilon = 0.01$)

(c) Recovered image ($\varepsilon = 0.001$)

(d) Recovered image ($\varepsilon = 0.0001$)

Figure 8. The detected image along with the recovered image using the Tikhonov regularization technique with varying choices of the regularization parameter $\varepsilon > 0$.

Variation of (4.13) yields $H[\delta f] = \delta v$. We introduce the following functional to be minimized

$$\inf \alpha\|v\|^2 + \beta\|f\|^2 + \|H[f] - f_{\text{obs}}\|^2, \quad \alpha, \beta > 0.$$

The variational approach yields

$$\alpha\langle v, \delta v\rangle + \beta\langle f, \delta f\rangle + \langle H^*(H[f] - f_{\text{obs}}), \delta f\rangle = 0. \tag{4.14}$$

We consider the adjoint equation

$$H^*[q] = p$$

and take an inner product of the adjoint equation by δf, $H[\delta f] = \delta v$ by q and obtain

$$\langle q, \delta v\rangle = \langle p, \delta f\rangle.$$

We compare with (4.14) and find that if

$$p = \beta f + H^*(H[f] - f_{\text{obs}}),$$

then we arrive at the control equation

$$\alpha v + q = 0.$$

We are in position now to state the following algorithm.

Algorithm We pick up the initial approximations $f_0 = f_{\text{obs}}$, $v_0 = 0$.
Step 1. Find f_n from the direct equation

$$H[f_n] = f_{\text{obs}} + v_n$$

Step 2. Find q_n from the adjoint equation

$$H^* q_n = \beta f_n + H^*(H[f_n] - f_{\text{obs}}).$$

Step 3. Find v_{n+1} from control equation:

$$v_{n+1} = v_n - \tau(\alpha v_n + q_n).$$

Parameter τ must be positive and less than $\frac{1}{\alpha}$.

We will carry out the computations according to this algorithm in future work and look forward to verifying that this control approach can lead to an accurate deblurring as an alternative to the deconvolution techniques explored in other parts of this report. We will further establish the rate of convergence of this method so as to compare with existing benchmarks provided by [5, 6, 11].

4.3 Wiener deconvolution

Contributed by Trishala Thakur, Tyler Fuller, Binan Gu

An additive noise n, independent of the signal f, as shown in (4.1), accounts for additional external influences on the image quality; for instance, images taken by space telescopes are almost always affected by atmospheric disturbances. More precisely, using the results from section 4.1, this noise-perturbed problem is given by

$$g = h * f + n \tag{4.1}$$

where n is a random variable with mean zero and finite variance, i.e.,

$$\mathbb{E}[n] = 0, \quad \sigma^2 := \mathbb{E}|n|^2 < \infty$$

Taking a Fourier transform of both sides, we obtain

$$\mathcal{F}[g] = \mathcal{F}[h]\,\mathcal{F}[f] + \mathcal{F}[n] \tag{4.2}$$

Because of the random nature of n, we cannot naively conclude that

$$f = \mathcal{F}^{-1}\left[\frac{\mathcal{F}[g] - \mathcal{F}[n]}{\mathcal{F}[h]}\right] \tag{4.3}$$

Instead, we propose the following mean square error minimization problem,

$$\min_{r \in L^2} \mathbb{E}|f - r * g|^2, \tag{4.4}$$

we find a minimizing r where $\widetilde{f} := r * g$ is considered as an approximation of f. Once we find r, we compute once again the Fourier transform

$$\mathcal{F}\left[\widetilde{f}\right] = \mathcal{F}[r * g] = \mathcal{F}[r]\,\mathcal{F}[g] \implies \widetilde{f} = \mathcal{F}^{-1}[\mathcal{F}[r]\,\mathcal{F}[g]]$$

and obtain the approximate solution $\widetilde{f}$.

To solve the minimization problem in (4.4), we appeal to the Plancherel theorem (or Parseval's identity); that is, the Fourier transform is an isometry with respect to the L^2-norm, and recast the problem in Fourier space,

$$\min_{r\in L^2} \mathbb{E}\,|f - r * g|^2 = \min_{R\in L^2} \mathbb{E}\,|F - RG|^2 \tag{4.5}$$

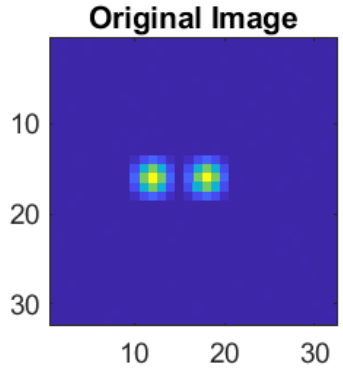
where the capitalized variables represent the Fourier transform of the lowercase ones. We readily observe that the recast minimization is a standard least square problem (see [3] for details on the solution technique).

The optimal least square approximate image $\widetilde{f}$ to (4.5) satisfies

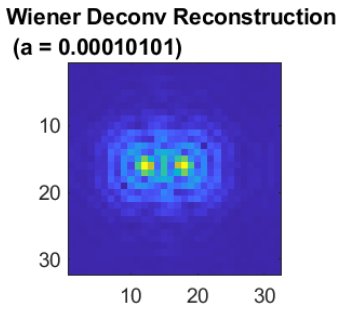
$$\mathcal{F}\left[\widetilde{f}\right] = \frac{\mathcal{F}[g]}{\mathcal{F}[h]} \frac{1}{1 + \frac{a^2}{|\mathcal{F}[h]|^2}} \tag{4.6}$$

where we have defined a hyper-parameter $a = \frac{\sigma^2}{\mathbb{E}|\mathcal{F}[f]|^2}$, known as the reciprocal of the signal-to-noise ratio. In practice, both the numerator and the denominator in a are unknown and thus need to be estimated based on a proposed model for the noise term n.

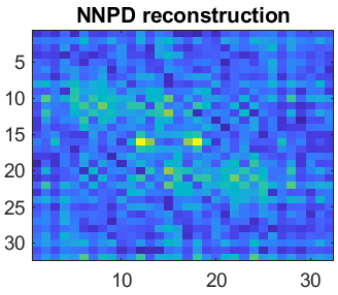
We tested various hyper-parameter values, and the results of these simulations are illustrated in Figure 9. There, we observe that Wiener deconvolution generally provides better results compared to the NNPD method in terms of sharpness of the object and the slightly reduced noise level caused by division by zero.



(a) Original Image (with noise)



(b) Reconstruction using Wiener deconvolution with optimal hyper-parameter a



(c) Reconstruction using NNPD

Figure 9. Comparison of different image reconstruction methods.

5 Smoothed point spread function (PSF)

This section proposes two modifications to the continuous analog of the NNPD: 1) the Truncated Gaussian Filter (TGF) and 2) the Super-Gaussian Optical Transfer Function (OTF). The benefit of considering continuous versions of the PSF is that the Fourier transform of the proposed filters can be computed analytically and thus directly numerically evaluated.

5.1 Truncated Gaussian filter

Contributed by Binan Gu and Burt Tilley

In this section, we study the continuous analog of the discrete PSF proposed in (1.4), with an additional consideration that the Gaussian filter is compactly supported due to the boundedness

of an image. More precisely, a Truncated Gaussian Filter (TGF) is a normalized Gaussian supported on a compact set in $\mathbb{R}^2$, typically a rectangle in image processing. We propose a PSF of the following form, supported on a centered rectangle of length and width a and b,

$$PSF\left(\boldsymbol{x}; a, b, c\right) = \alpha\left(c\right) e^{-c|\boldsymbol{x}|^2} \mathbf{1}_{\{-a \le x \le a, -b \le y \le b\}}, \tag{5.1}$$

where the normalizing factor satisfies

$$\alpha\left(c\right) = \frac{c}{\pi} \frac{1}{\operatorname{erf}\left(\sqrt{c}a\right) \operatorname{erf}\left(\sqrt{c}b\right)},$$

and c is the inverse width of the Gaussian and $\operatorname{erf}\left(x\right)$ is the one-dimensional error function, given by

$$\operatorname{erf}\left(x\right) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

One may assume $a = b$ for a simpler adaptation to square images, but we keep a and b as general tunable parameters. The Optical Transfer Function (OTF), obtained by taking a Fourier transform of this PSF (viewed as the product of a Gaussian and an indicator function), is the convolution of a Gaussian and the sinc function, i.e.,

$$OTF\left(\xi_1, \xi_2\right) = \alpha\left(c\right) \mathcal{F}\left\{e^{-c|\boldsymbol{x}|^2}\right\} * \mathcal{F}\left\{\mathbf{1}_{\{-a \le x \le a, -b \le y \le b\}}\right\} \tag{5.2a}$$

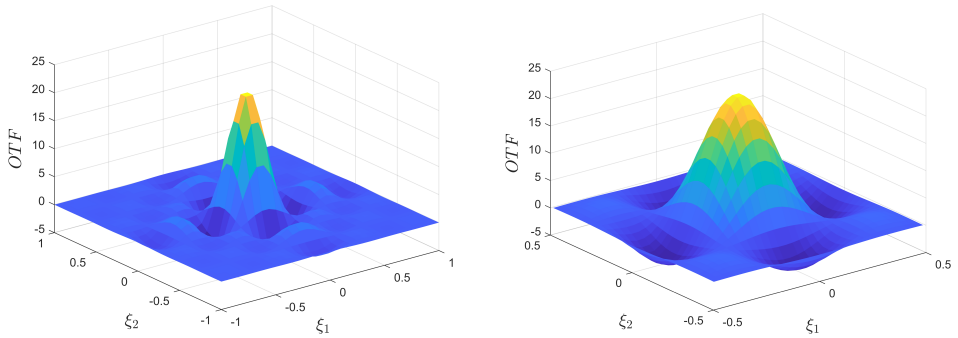
$$= \alpha\left(c\right) \left(\frac{\pi}{c} e^{-\frac{\pi^2\left(\xi_1^2 + \xi_2^2\right)}{c}}\right) * \left(\frac{1}{\pi^2} \frac{\sin\left(2\pi a \xi_1\right) \sin\left(2\pi b \xi_2\right)}{\xi_1 \xi_2}\right) \tag{5.2b}$$

Analytically, the 2D Fourier transforms and their convolutions in eq. 5.2b are supported on the entire $\mathbb{R}^2$ but must be truncated for computation. We set the frequency variables $(\xi_1, \xi_2) \in [-1, 1]^2$ (based on the Nyquist frequency), with a common mesh size $\Delta\xi$ for each dimension, with $\Delta\xi$ small enough such that the grid is $kM \times kN$ for $k > 1$ (where $k = 1$ corresponds to the dimension of the original detected image). We use the middle $M \times N$ grid points in frequency space (to match the image size) to perform the deconvolution step

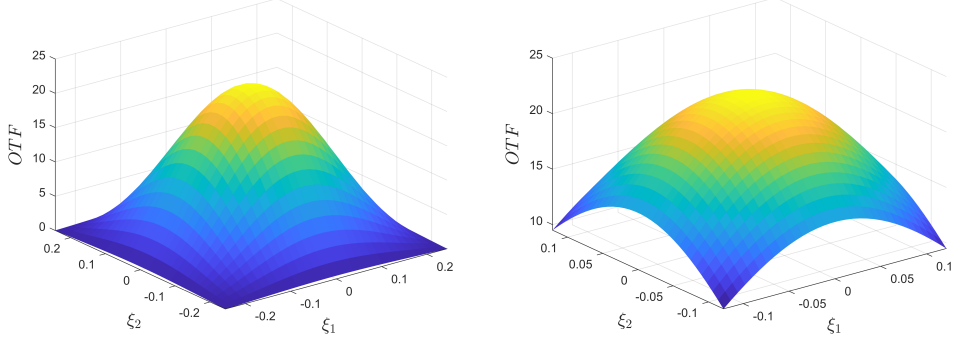
$$f\left(x_1, x_2\right) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\left[g\right]\left(\xi_1, \xi_2\right)}{OTF\left(\xi_1, \xi_2\right)}\right\}.$$

In Figures. 10a-10d, we provide a few examples of the Truncated OTF at various resolutions, with an intended image of size 32-by-32. Provided a frequency domain of $(\xi_1, \xi_2) \in [-1, 1]^2$, $\Delta\xi$ must be smaller than $\frac{1}{16}$ so that the 32-by-32 image fits the meshgrids (exactly with $\Delta\xi = \frac{1}{16}$). If a smaller $\Delta\xi$ is chosen, then the image domain is strictly embedded in the interior of the frequency domain. Figure. 10a uses a coarse mesh size of $\Delta\xi = \frac{1}{16}$, producing a truncated OTF showing both the Gaussian peak at the origin and the sinc wave, admitting several zeros away from the origin. In Figure. 10b, we use $\Delta\xi = \frac{1}{32}$, a mesh size twice as fine, which effectively embeds the image on $\left[-\frac{1}{2}, \frac{1}{2}\right]^2$ (see the new limits of the axes). The zeros are still present near the origin. As we decrease $\Delta\xi$ through Figures. 10c-10d, the zeros disappear as the truncated OTF has more Gaussian features than sinc ones. For larger images (more pixels), we may increase a and b to accommodate for the expanded real domain.

The convolution in the OTF is computed by a Gauss-Hermite quadrature with three query points. We have four parameters, a, b, and c, for the TGF and the mesh size $\Delta\xi$ that determines the resolution of the OTF in frequency space. The TGF parameters may be tuned so that a combined metric involving the relative L^2-error between the detected and filtered image, signal-to-noise ratio, and sharpness is optimized, an idea to be considered in future work. Furthermore, the peak of the Gaussian can be tuned via c, which can be inferred from the measurement values of the discrete Gaussian used in the NNPD.



(a) $\Delta\xi = \frac{1}{16}$ (b) $\Delta\xi = \frac{1}{32}$



(c) $\Delta\xi = \frac{1}{64}$ (d) $\Delta\xi = \frac{1}{128}$

Figure 10. Truncated OTF per eq. 5.2b for an intended 32-by-32 image at various resolutions of the mesh grid: (a) $\Delta\xi = \frac{1}{16}$ (b) $\Delta\xi = \frac{1}{32}$ (c) $\Delta\xi = \frac{1}{64}$ and (d) $\Delta\xi = \frac{1}{128}$. For all sample plots, Gaussian parameters per eq. 5.1 are fixed: $a = b = 2$ and $c = 15$.

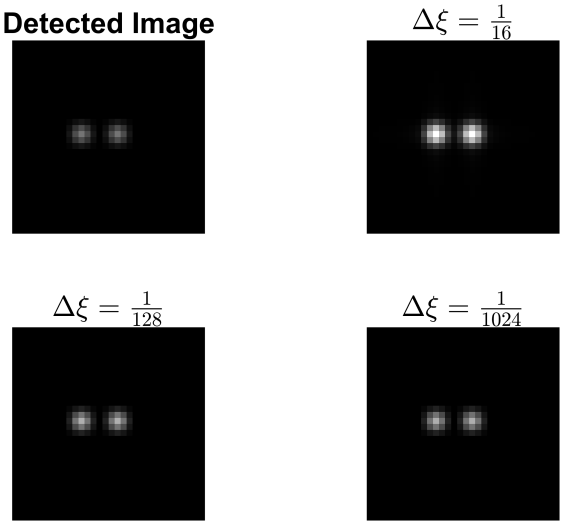


Figure 11. TGF with parameters $a = b = 0.25$, $c = 15$

In Figures. 11-14, the TGF parameter values are chosen so that the improvement across various values of $\Delta\xi$ is visible. In particular, figure. 12 demonstrates the effect of under meshing at $\Delta\xi = \frac{1}{128}$ that barely preserves the structure of the original image, but the improvement is stark at finer mesh sizes. A similar effect can be seen in figure. 13 where $\Delta\xi = \frac{1}{512}$ is considered too small because the image size has increased from figure. 12.

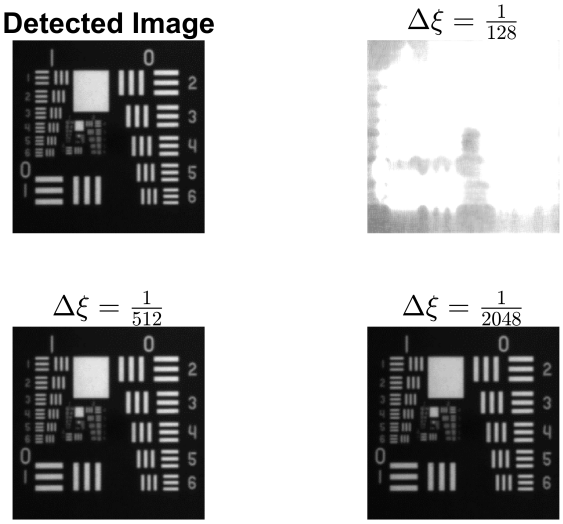


Figure 12. $a = b = 0.5$, $c = 15$, image size 246-by-245.

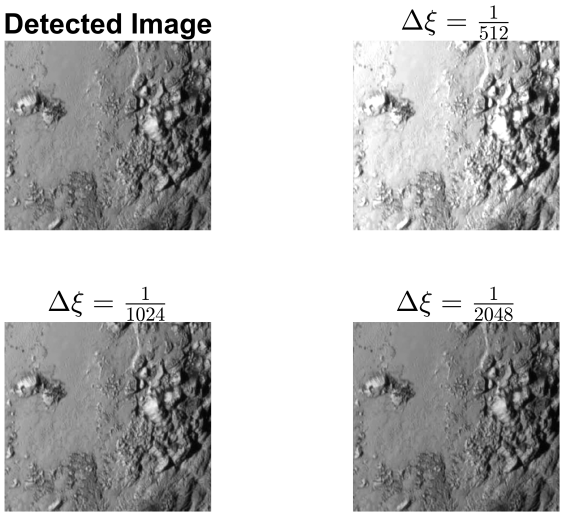


Figure 13. Same TGF parameters as Figure 12, image size 441-by-481.

5.2 Regularization of the optical transfer function (OTF)

Contributed by Fei Cao

Recall the formula (4.5) for the Fourier transform of the original image:

$$\mathcal{F}[f](\xi,\eta) = \frac{\mathcal{F}[g](\xi,\eta)}{\mathcal{F}[h](\xi,\eta)}. \tag{5.3}$$

To overcome the possibility of a (possibly) vanishing OTF in the denominator, we can also try to *r*egularizing the OTF directly. For instance, one can employ a parametric "Gaussian-like" OTF,

$$\mathcal{F}[h](\xi,\eta) = \mathrm{e}^{-\xi^\gamma - \eta^\gamma} \tag{5.4}$$

with $\gamma \in \mathbb{R}$ being a tunable parameter or an (inverse) polynomial-type OTF of the form

$$\mathcal{F}[h](\xi,\eta) = \frac{1}{1 + \gamma_1\,\xi^{\gamma_2} + \gamma_1\,\eta^{\gamma_2}} \tag{5.5}$$

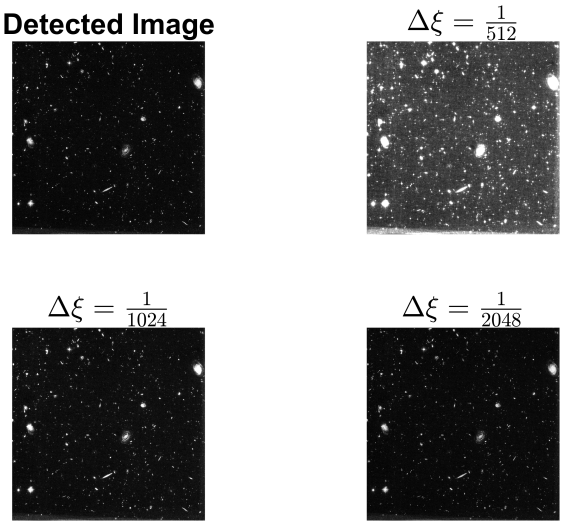


Figure 14. Same TGF parameters as Figure 12, image size 820-by-820.

in which γ_1 and γ_2 are tunable parameters. Numerical experiments suggest that a marginal improvement over the classical benchmark can be achieved after careful tuning of hyper-parameters.

Figure 15. **Left:** The recovered image using the NNPD OTF (which consists of linear combination of sin and cos functions); **Right:** The recovered image using the smoothed OTF of the form (5.5) with $\gamma_1 = 1$ and $\gamma_2 = 0.01$.

6 Sparse deblur dictionary learning

Contributed by William Thompson, Tyler Fuller, Clayton Coonrod

During the workshop, we applied a non-classical method of debluring that was based on the reconstruction of the original image using some over-complete set, aka dictionary.

The article titled "Direct Sparse Deblurring" by Yifei Lou, Andrea L. Bertozzi, and Stefano Soatto developed a deblurring algorithm that leverages the sparse characteristics of natural images, avoiding the numerical challenges associated with traditional methods that solve ill-posed inverse problems. Deblurring aims to reverse the effects of blurring caused by convolution with a known kernel (that might have some parameters to fit). The proposed method operates through the following steps: creating an over-complete basis (dictionary) for the original image, performing sparse coding on the blurred image using the blurred version of the dictionary, and reconstructing the deblurred image using the coefficients obtained from sparse coding. This

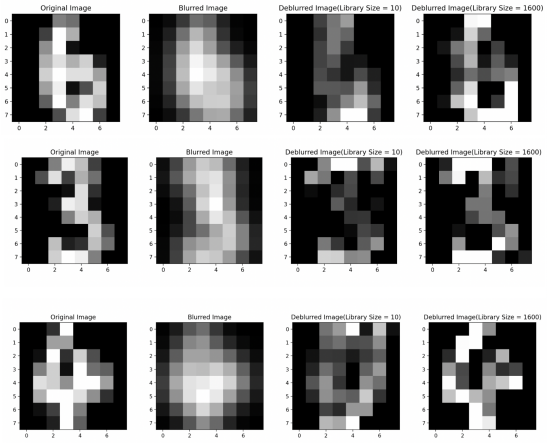


Figure 16

approach avoids solving an inverse problem directly, instead focusing on a generative model approach, sometimes referred to as "analysis by synthesis".

An image patch is assumed to be represented sparsely in a chosen dictionary. The blurred image can be expressed as a combination of blurred basis elements with the same sparse coefficients that represent the original image. This forms the basis of the direct sparse deblurring method. The deblurring algorithm minimizes the sparse representation error for all local patches. The optimization involves solving for the sparse coefficients that best approximate the blurred image with respect to the blurred dictionary. The reconstructed image is then obtained by combining these coefficients with the original (non-blurred) dictionary elements.

In practical implementation, the image is divided into patches to manage computational complexity. A refined approach ensures that the blurred and clear basis elements are consistent in their domain sizes to address boundary effects. The algorithm includes steps for solving the coefficients from the measured image and stitching the patches to form the deblurred image. Weighted averaging is used to combine overlapping patches, with weights chosen based on the sparsity of their representation, promoting better reconstruction quality.

The effectiveness of the method depends on the coherence of the dictionary. Highly coherent dictionaries make distinguishing between similar atoms difficult for the sparse coding algorithm, affecting the deblurring quality. The averaging process for overlapping patches can degrade the reconstruction quality. Future work may focus on improving this aspect by using partitioning strategies to minimize boundary artifacts.

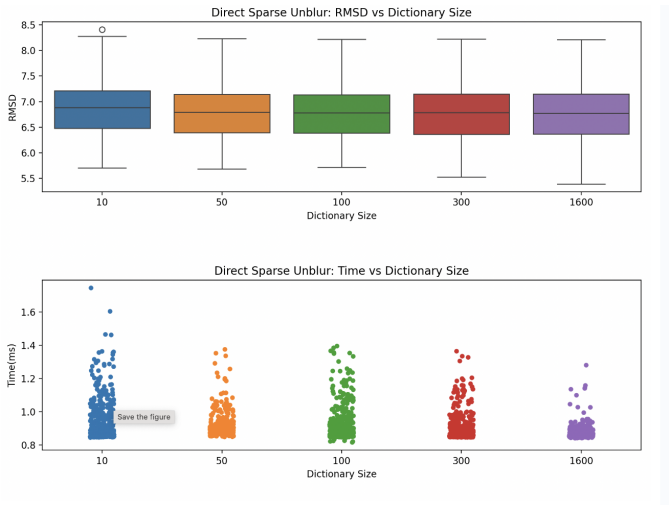


Figure 17

7 Conclusion and discussions

7.1 Summary of Results

Image deblurring is a classical problem still being studied by modern approaches. In this report, we explored several methods based on Laplace transform and its inversion technique, Cauchy integration via a translational basis and Fourier Transform, regularized optimization, stochastic optimization, optimal control, point spread function smoothing, and dictionary learning. Among these approaches, direct inversion methods such as the Weeks' method (one of the Laplace transform-based methods) and Cauchy integration assume either the original signal or the point spread function has an underlying basis whose linearity structure can aid in the inverse transform. The fixed Talbot's method works directly with contour choices in contour integration as a numerical inversion method. Indirect inversion approaches, such as regularized optimization, the optimal control approach, and the Wiener deconvolution, recast the deblurring as a minimization problem that utilizes the rich theory of constrained optimization; however, these methods still suffer from the introduction of hyperparameters, which require further tuning in practice. Furthermore, the original discrete NNPD approach inspired us to work directly in the Fourier space, replacing the Optical Transfer Function (the Fourier transform of the PSF) with either a continuous kernel truncated at the image boundary or a modified "Super-Gaussian"-like kernel with a couple of added parameters. All methods in this report have shown promising improvement in sharpness and signal-to-noise ratio, reduced computational cost, and improved accuracy, providing fertile ground for future directions.

7.2 Future Directions

Many great ideas did not come to fruition within the workshop's short time frame but are nevertheless potential future directions to pursue. Some of the proposed ideas are already listed in the individual sections above. We discuss a few key ideas that are realizable in the short term.

The fixed Talbot's method (per section 2.2) has proven to be a robust numerical inversion of the Laplace transform in the test examples. At the same time, the only drawback occurs when the integrand contains undesirable branch cuts, which can be ameliorated by selecting an appropriate basis for the signal during the forward transform. Secondly, the control approach

(per section 4.2) is a more general problem than the image deblurring, which may give rise to more extensive general results that allow more explorations in various forms of the point spread function. Lastly, as implemented in this report, the OTF smoothing (per section 5.1) operates on the entire image domain with a few universal parameters. Instead, one may choose these parameters based on the local information and patch together the image across different parts of the image, a process readily parallelizable without incurring additional computational time. We look forward to carrying out these directions with each subgroup's faculty members and graduate students.

References

[1] Evans, G. A. (1993). Numerical inversion of laplace transforms using optimal contours in the complex plane. *Int. j. comput. math.*, **49**, 93–105.

[2] Evans, G. A., & Chung, K. C. (2000). Laplace transforms inversion using optimal contours in the complex plane. *Int. j. comput. math.*, **73**, 531–543.

[3] Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing.* Pearson. Chap. 5.

[4] Greengard, L., & Lee, J.-Y. (2004). Accelerating the nonuniform fast fourier transform. *Siam review*, **6**(3), 443–454.

[5] Kano, P. O., Brio, M., & Moloney, J. V. (2005). Application of weeks method for the numerical inversion of the laplace transform to the matrix exponential. *Comm. math. sci.*, **3**(3), 335–372.

[6] Kano, P. O., Brio, M., Dostert, P., & Cain, J. (2012). Dempster–shafer evidential theory for the automated selection of parameters for talbot's method contours and application to matrix exponentiation. *Comput. math. appl.*, **63**(11), 1519–1535.

[7] Murli, A., & Rizzardi, M. (1990). Algorithm 682. talbot's method for the laplace inversion problem. *Acm trans. math. softw*, **16**, 158–168.

[8] Scherzer, Otmar. (2010). *Handbook of mathematical methods in imaging.* Springer Science & Business Media.

[9] Talbot, A. (1979). The accurate numerical inversion of laplace transforms. *J. inst. math. appl.*, **23**(3), 97–120.

[10] Valkó, P. P., & Abate, J. (2003). Comparison of sequence accelerators for the gaver method of numerical laplace transform inversion. *Comput. math. appl.*, **48**, 629–636.

[11] Wang, Y., & Lu, Y. (2019). Enhance image resolution with nearest neighbor pixel deconvolution. *Page 109980S of:* Vizgaitis, Jay N., Marasco, Peter L., & Sanghera, Jasbinder S. (eds), *Advanced optics for imaging applications: Uv through lwir iv*, vol. 10998. SPIE.

[12] Weideman, J.A.C. (1999). Numerical inversion of the laplace transform using the talbot contour: a survey and comparison. *Math. comp.*, **69**(231), 501–536.