

Uncertainty Propagation in Regularization-Based Image Deblurring

Research Project Proposal

Thomas Pasfield

As submitted on:
October 28, 2024

Supervising Professor:
Dr. Harihar Khanal

Department of Mathematics
Embry-Riddle Aeronautical University
Daytona Beach, FL

Introduction

Image blur hinders quantitative image analysis, especially in the field of radiographic imaging. By approaching image deblurring as deconvolution (an ill-posed inverse problem), methods such as Tikhonov and Total Variation Regularization present appropriate solutions. As non-blind deblurring methods, the regularization methods require an estimate or exact measurement of the Point Spread Function (PSF) of the optical system. The PSF is unknown but can be estimated from a calibration image. This estimation process introduces uncertainty into the final solution. In this work, I aim to provide a comprehensive analysis of the uncertainty propagation from the PSF estimate through the regularization processes. The results will provide insight into the deblurring methods already in use, and allow for informed analysis of the deblurred outputs.

Existing work in this application is thoroughly explored by Dr. Jordan Pillow of the Nevada National Security Site in his 2021 PhD Thesis. [2]

Problem Description

Blur as Convolution

Performing regularization requires this problem to be approached with traditional computer vision techniques rather than the machine learning techniques that have become more popular. These traditional computer vision techniques often represent operations on images as linear models, which is a practice which will be continued here. Our image, \mathbf{X} , is a two-dimensional matrix representing a black-and-white image or a single color channel from a color image, with shape (m, n) . The color depth of the image is largely irrelevant. The process of image blurring can be represented as a convolution of the PSF across the image, cropped to the original window of the image. The blurred image is represented by \mathbf{B} , which shares the same shape (m, n) of the original image. This system is shown in Equation (1).

$$\mathbf{B} = \mathbf{X} * \text{PSF} \quad (1)$$

Creating a Linear Model

This equation, however, is not a linear model and cannot be used in regularization. It must first be discretized into matrices. When the PSF depends only on the X and Y axes, it can be known as separable blur. Separable blur can be represented by two different convolution operations: one where the kernel only affects the X-axis, and another where the kernel only affects the Y-axis. A 1-D PSF can be represented as a vector along the axis that is being blurred along. For a rather simple system, this 1-D PSF may approximate a Gaussian distribution. This is demonstrated within Figure 1. From this point on, the exact structure used to discretize the convolution is dependent on the boundary conditions applied to the image. In this case, it is certain that there

is a zero boundary, so all work from this point forward will use the structures and methods specific to that case. See *Deblurring Images: Matrices, Spectra, and Filtering* [1] for more information on this structure.



Figure 1: A plot of a Gaussian distribution PSF and a representation of a corresponding vector to convolve this PSF along the x-axis.

Convolutions can be discretized quite easily by taking a 1-D kernel and creating a Toeplitz matrix with it. This matrix, when multiplied with the vectorized (column-stacked) image vector, performs a convolution of the kernel along the expected axis. This discretized convolution can be seen in Figure 2. More work is required to discretize a 2-D convolution. First, discrete convolution matrices must be generated for both the blur along the x and y axes. Then the 2-D discrete convolution matrix is formed by taking the Kronecker product of the two components. This 2-D convolution matrix is shown within Figure 3. This 2-D convolution matrix will be represented via \mathbf{A} , and its respective components by \mathbf{A}_x and \mathbf{A}_y . This new matrix \mathbf{A} has a size of (mn, mn) . The convolution is performed by multiplying the convolution matrix with the column-stacked vectorized image matrix. These relationships are all observed in Equations (2) through (4).

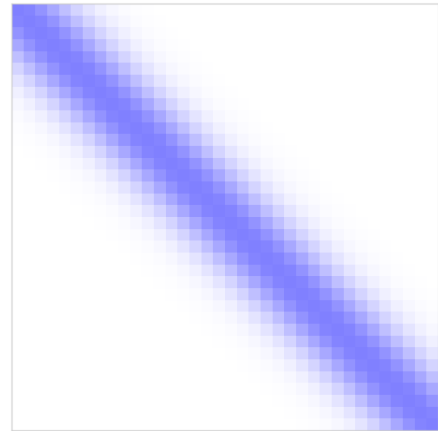


Figure 2: The 1-D PSF extrapolated into a Toeplitz matrix. This matrix now functions as a convolution along the x-axis when multiplied with a vectorized image.

$$\mathbf{A} = \mathbf{A}_x \otimes \mathbf{A}_y \quad (2)$$

$$\mathbf{B} = \mathbf{A}_y \mathbf{X} \mathbf{A}_x^T \quad (3)$$

$$\mathbf{b} = \text{vec}(\mathbf{B})$$

$$\mathbf{x} = \text{vec}(\mathbf{X})$$

$$\mathbf{b} = \mathbf{A} \mathbf{x} \quad (4)$$

Now that we have these components defined, we can assemble a linear model for an image, seen in Equation (5). This linear model represents a blurred image as a convolution of the PSF across the true image, along with added noise. This added noise can come from various sources, but cannot be modeled on its own. The noise is represented by ϵ , and it takes the shape of a $(1, mn)$ vector of normally distribution values. The presence of this noise makes this problem an ill-posed, inverse problem, which invites the regularization methods to follow.

$$\mathbf{b} = \mathbf{A} \mathbf{x} + \epsilon \quad (5)$$

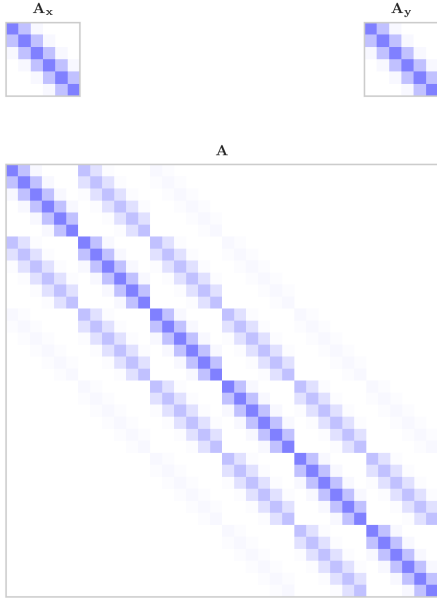


Figure 3: A 2-D discrete convolution matrix formed by performing the Kronecker product of the 1-D discrete convolution matrices.

Regularization

Regularization is the standard approach to finding plausible solutions for ill-posed inverse problems, and the techniques and strategies are thoroughly described in *Computational Methods for Inverse Problems* [3]. Regularization biases an optimization process used to find a possible solution. This bias allows one to tune the output towards an expected property of the solution, such as smoothness.

The two forms of regularization which this work will explore are Tikhonov Regularization, which is also known as L2 or Ridge Regression, and Total Variation Regularization. Tikhonov regularization biases the optimization towards smoother results, while total variation regularization biases results towards blocky results.

Tikhonov Regularization

To bias the optimization towards smooth results, the penalty function attempts to minimize the residuals between the blurred estimate of the solution and the blurred image. This is just a least-squares regression and ensures that the solution maintains some semblance of the blurred image. To make this Tikhonov Regularization, the variance is added to the penalty function, which reduces the cost of smoother results compared to noisy or blocky ones. The full optimizer is Equation (6). The addition of the Tikhonov factor improves the conditioning of the system, allowing for an explicit form shown in Equation (7). Matrix \mathbf{L} is the Tikhonov Matrix, and is simply the identity matrix in this application, while α is a weight parameter that allows for control over how much penalty the variance provides.

$$\mathbf{x}_{TIK} = \text{argmin}_x (||\mathbf{A} \mathbf{x} - \mathbf{b}||_2^2 + \alpha ||\mathbf{L} \mathbf{x}||_2^2) \quad (6)$$

$$= (\mathbf{A}^T \mathbf{A} + \alpha \mathbf{L}^T \mathbf{L})^{-1} \mathbf{A}^T \mathbf{b} \quad (7)$$

Total Variation Regularization

To bias a solution towards blocky or segmented results, the penalty function must be changed. The base function is still a least-squares solver, as we still need to ensure that the blurred true image is as close to the actual blurred image as possible. Added to the penalty is the weighted Total Variation. This is found using a matrix 1-norm of the product of a discrete derivative operator \mathbf{D} with the current image solution. The full optimizer is below in Equation (8). α shares the same use here as it does with the Tikhonov regularizer.

$$\mathbf{x}_{TV} = \text{argmin}_x \left(\frac{1}{2} ||\mathbf{A} \mathbf{x} - \mathbf{b}||_2^2 + \alpha ||\mathbf{D} \mathbf{x}||_1 \right) \quad (8)$$

Parameter Selection

You may notice that both of these regularization optimizers require a weight parameter α . This can be selected arbitrarily by whoever executes the code based on observed changes or prior knowledge, or it can be estimated algorithmically. Algorithmic determination is better suited for this application, though may yield worse results than the user may want, due to ideal outcomes being subjective in nature and difficult to quantify in an optimizer.

Ideally, we could just compare the outcome with an arbitrary parameter to the true image we are solving for, then vary the parameter to minimize residuals. Due to obvious reasons, this is an unrealistic case. We must instead find a

reasonably accurate method to predict the error in the solution. This is done via Generalized Cross Validation (GCV).

Unlike typical GCV use, we don't have knowledge of the true image to validate against, but this is easy to work around. We know the blurred image, so we simply blur our solution and compare the blurred solution to the known blurred image. Curtis Vogel [3] provides a GCV cost function for Tikhonov regularization, provided in Equation (9), where n is dimension stated earlier, and \mathbf{F} is an influence matrix defined in Equation (10). The book provides no derivation of the influence matrix for total variation regularization.

$$\text{GCV}(\alpha) = \frac{\frac{1}{n} \|\mathbf{r}_\alpha\|^2}{\left[\frac{1}{n} \text{trace}(\mathbf{I} - \mathbf{F}_\alpha)\right]^2} \quad (9)$$

$$\mathbf{F}_\alpha = \mathbf{A} (\mathbf{A}^T \mathbf{A} + \alpha \mathbf{L}) \mathbf{A}^T \quad (10)$$

Though there is no derivation for total variation, there are other works which attempt to create a strong GCV cost function for it. Dr. You-Wei Wen of Hunan Normal University [4] derives a GCV function for total variation by finding a representation resembling Tikhonov regularization, then using the known Tikhonov derivation to find the cost. This method will likely be implemented in this work, but that is uncertain.

Methodology

Now that we have the foundation of this work laid out, the actual focus of what will be executed must be explained. Within the regularization systems described, all systems rely heavily on the blurring matrix \mathbf{A} . If you recall, this matrix is a prediction of what the real blurring process changed, and not a perfect match. This prediction relies on assumptions which may be flawed, and the actual model matching is influenced by noise and randomness within a given observation. We would like to understand how error in this prediction propagates through the regularization and GCV processes to affect the final resolved image.

The calibration process used to predict the blurring matrix is rather simple. To observe blur, we can simply image a line pair (Figure 4) and measure the value across it, as seen in Figure 5. These measurements describe the blur directly, though the noise makes the raw data unsuitable for further work. The PSF must be smooth and continuous, so we must find a smooth, continuous function to match the data. The curve clearly approximates a logistic function, so that is what is used in these examples. Figure 6 shows how the derivative of this function has the shape of a normal distribution, and thus fits the processes seen in Figure 1. This curve is prepared for the process by simply truncating it to halfway, and padding the end with zeros to achieve the required size, seen in Figure 7. In an ideal scenario, there would be no blur at all and the edge of the line pair would simply be a jump discontinuity to the new value.

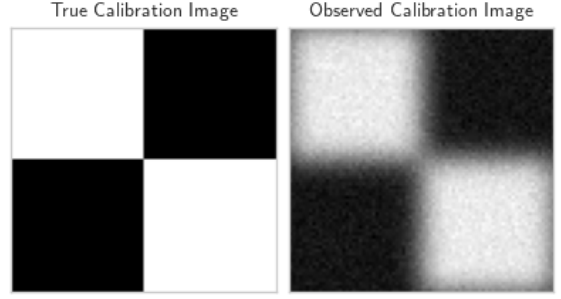


Figure 4: A side-by-side comparison of the true line pair calibration image, containing jump discontinuities from high to low values along both the x and y axes, with the observed result. Note that the observed result is blurred, has considerable noise, and has lost contrast.

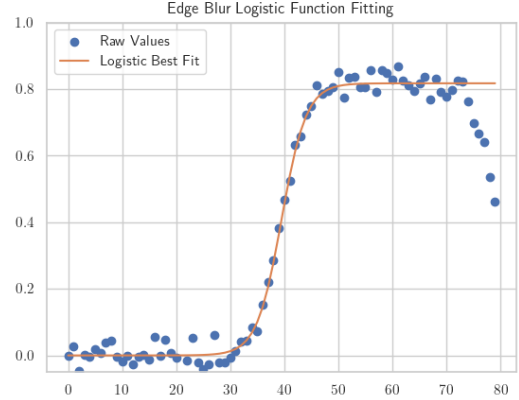


Figure 5: The values across a single row of pixels from the blurred image seen in Figure 4. The raw values are noisy but follow a logistic function, so a logistic function has been fit to the data.

The noise alters the residuals when performing the curve fitting, leading to uncertainty in what the real blur is. Due to this, one can curve fit in a manner which gives a lower and upper bound on values which produce a solution with the residual beneath some arbitrary threshold. Sampling can be performed within that range, and the resulting curve can be used for the regularization process. There may be a way to use the Signal-To-Noise ratio to be able to quantify the confidence of how well the curve fit may match the real blur performed.

This problem takes the form of forward propagation of uncertainty, and will consist of taking samples and propagating them through the models to compare the results. Ironically, such a complex foundation of the work has ended up resulting in a rather simple practice in descriptive statistics in the end.

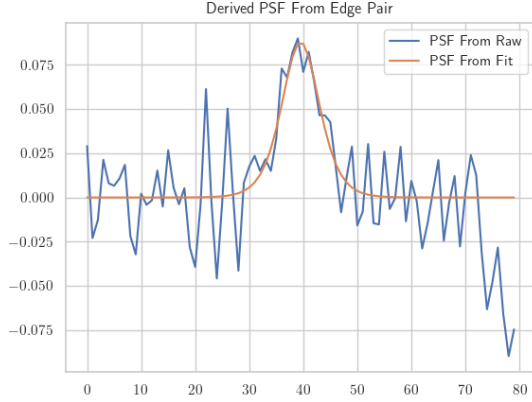


Figure 6: The discrete derivative values of both the raw values and the logistic best fit, showing a smooth continuous normal-like function from the logistic curve, and effectively random noise from the raw values. This highlights the importance of the curve fit.

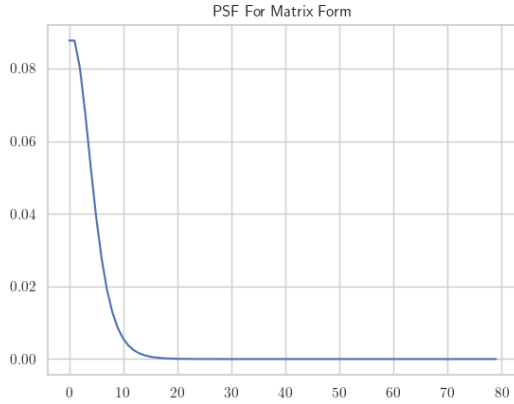


Figure 7: The truncated derivative curve, normalized to a sum of 0.5, such that it can be used to create the blurring matrix in the process initially described in Figure 1.

To conduct this forward propagation, curve fitting will be performed to determine the best fit. The curve will fit to several rows of the calibration image, rather than just one, to attempt to reduce the effect of noise. Based on the initial solution, Monte Carlo sampling allows for sampling around the parameter. Normally-distributed samples should work well. Each sample will be processed, and the discrete convolution matrix generated.

Initial plans for the computation consist of the following:

1. Assign a sampled parameter to two threads from a set of worker threads.
2. Each parameter is run through the Tikhonov and Total Variation regularizers.

- Tikhonov regularization is fairly easy to accelerate with Numexpr or Numba.
- Total Variation is more complex, but can be distributed with PyLOPs, and PyLOPs-MPI.

3. Results are aggregated, new workers are deployed until all samples return results.
4. Solution images will be stacked into a tensor, with statistics computed along the new axis element-wise, such as standard deviation.

Deliverables

- ~~Topic Submission~~ Due September 20th
- ~~Project Proposal~~ Due October 28th
- Preliminary Report/Presentation Due November 18th
- Rough Draft Due December 2nd
- Final Report Due December 9th

References

- [1] Per Christian Hansen, James G. Nagy, and Dianne P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering*. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics, Jan. 2006. 129 pp. ISBN: 978-0-89871-618-4. DOI: 10.1137/1.9780898718874. URL: <https://epubs.siam.org/doi/book/10.1137/1.9780898718874> (visited on 10/23/2024).
- [2] Jordan A. Pillow et al. "Bayesian Spatially Varying Multi-Regularization Image Deblurring". In: *Inverse Problems in Science and Engineering* - <https://www.tandfonline.com/journals/gipe20> (Dec. 1, 2021). Institution: Nevada National Security Site/Mission Support and Test Services LLC (NNSS/MSTS), North Las Vegas, NV (United States) Number: DOE/NV/03624-1256. URL: <https://www.osti.gov/biblio/1835841> (visited on 10/23/2024).
- [3] Curtis R. Vogel. *Computational Methods for Inverse Problems*. Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, Jan. 2002. 195 pp. ISBN: 978-0-89871-550-7. DOI: 10.1137/1.9780898717570. URL: <https://epubs.siam.org/doi/book/10.1137/1.9780898717570> (visited on 10/23/2024).
- [4] You-Wei Wen et al. "Using generalized cross validation to select regularization parameter for total variation regularization problems". In: *Inverse Problems & Imaging* 12.5 (2018), pp. 1103–1120. ISSN: 1930-8345. DOI: 10.3934/ipi.2018046. URL: <http://aimsciences.org/article/doi/10.3934/ipi.2018046> (visited on 10/28/2024).