

# Extending the Lifetime of Broken Mobile Phone Lenses Through Deconvolution

Felipe Zanforlin Mautner  
Carnegie Mellon University  
Pittsburgh, PA, United States  
fmautner@andrew.cmu.edu

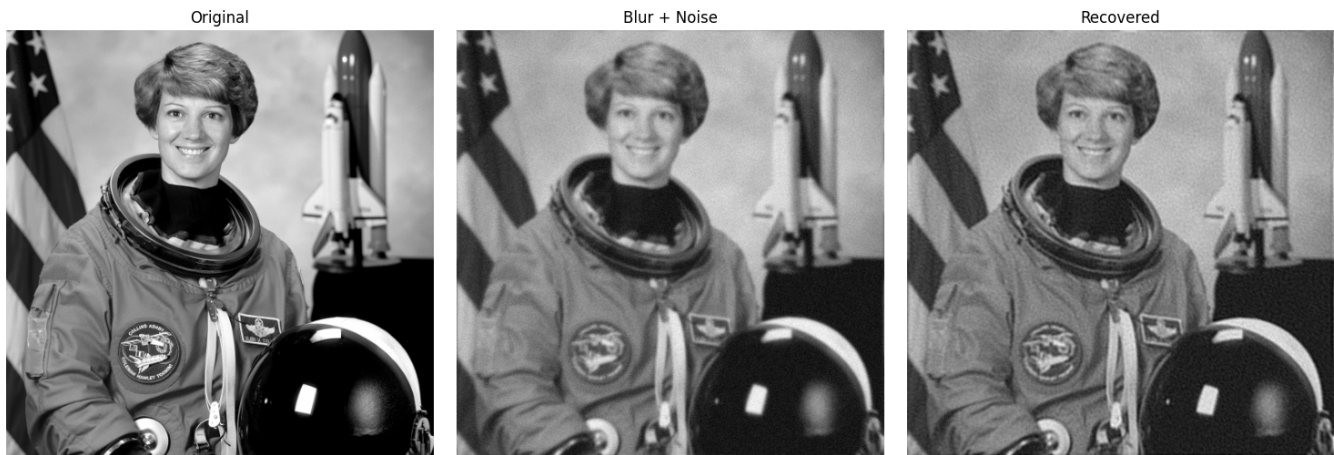


Figure 1: Deconvolution results with DR-TV Method.

## ABSTRACT

Smartphone camera lenses can crack at seemingly random times and for no good reason, and repairing these cameras through official channels can be prohibitively expensive<sup>1</sup>. These cracks can turn sophisticated lenses into single element simple lenses, resulting in blurry images that render the camera essentially unusable. This project implements and compares different deconvolution techniques to retrieve high quality images from damaged lenses.

## KEYWORDS

Aberration, Fourier Transform, Deconvolution, Point Spread Function (PSF), Alternating-Direction Method of Multipliers (ADMM), Douglas-Rachford Splitting.

## 1 INTRODUCTION

Modern lens designs employ multiple optical elements that work in concert to minimize aberrations inherent to spherical lenses. These aberrations include chromatic aberration, where different wavelengths of light focus at different distances, and spherical aberration, where light rays passing through different zones of the lens focus at different points. While high-end photographic lenses use numerous elements to correct these issues, they consequently become bulky and expensive.

Mobile phone cameras represent a different design paradigm, where space and cost constraints necessitate simpler lens constructions. While these lenses have seen remarkable improvements in

quality over the past decade, their simplified design and exposed position make them particularly vulnerable to damage. When a phone's lens is cracked or scratched, the carefully engineered multi-element system can effectively degrade into a simple single-element lens, introducing significant aberrations. These aberrations typically manifest non-uniformly across the image, resulting in a spatially-variant Point Spread Function (PSF), where different regions of the image exhibit distinct blur patterns based on their position relative to the optical axis.

Recent work in computational deblurring techniques have shown great promise in retrieving high quality images from such undesirable lens conditions. This project surveys three of them, mainly, Wiener Deconvolution, Total Variation (TV) Deconvolution with ADMM for constant PSF, and TV Deconvolution with the Douglas-Rachford algorithm for space-varying PSFs.

## 2 RELATED WORK

Image deconvolution has been an active area of research for several decades, with applications ranging from astronomical imaging to computational photography. Here we review key developments relevant to our work on recovering images from damaged phone lenses.

### 2.1 Classical Deconvolution

The foundation of image deconvolution lies in classical techniques such as Wiener filtering, which provides optimal linear filtering in the presence of additive noise. While computationally efficient, these methods often produce ringing artifacts and struggle with

<sup>1</sup>See Apple's repair cost estimates for different iPhone models <https://support.apple.com/iphone/repair>

spatially-varying blur. Richardson-Lucy deconvolution, originally developed for astronomical imaging, offers an alternative iterative approach but similarly suffers from noise amplification and boundary artifacts [7].

## 2.2 Optimization-Based Approaches

Modern approaches frame deconvolution as an optimization problem, typically incorporating image priors to improve robustness. These optimization problems often have dual formulations that have been extensively studied in the literature. The Alternating Direction Method of Multipliers (ADMM), originally introduced by [3] and re-surfaced by [1], has emerged as a powerful framework for solving these problems efficiently.

Chambolle and Pock [2] introduced a first-order primal-dual algorithm that provides an alternative optimization framework, offering theoretical guarantees and practical efficiency for imaging problems. Their work has been influential in developing algorithms for various image processing tasks, including deconvolution with multi-channel priors.

## 2.3 Spatially-Varying Deconvolution

The challenge of spatially-varying PSFs has been addressed by several researchers. O'Connor and Vandenberghe [6] presented a comprehensive approach using the Douglas-Rachford algorithm, demonstrating its effectiveness for space-varying kernels. Their method is particularly relevant for our work as it handles the varying blur patterns typical of damaged lenses.

## 2.4 Computational Imaging through Simple Lenses

Heide et al. [4] extend on Optimization-Based Approaches and Spatially-Varying Deconvolution by including cross-channel loss terms to account for chromatic aberration, based on the First-Order Primal-Dual algorithm introduced by Chambolle and Pock[2].

# 3 BACKGROUND

## 3.1 Imaging Model

The image capturing process is modeled as a convolution of the ground truth image  $i$  with a PSF kernel  $k$ , usually modeled as a circular low-pass filter, with additive noise  $n$  due to in-camera processes such as the conversion of the original analog signal to digital data. name the resulting blurry image  $b$ :

$$b = i * k + n$$

This can be equivalently expressed as a matrix-vector problem by letting  $\mathbf{b}$ ,  $\mathbf{i}$  and  $\mathbf{k}$  be the vectorized forms of  $b$ ,  $i$  and  $k$  respectively. The convolution operation can then be expressed as multiplication with a matrix  $K$  constructed from the kernel  $k$  where row of  $K$  represents the kernel centered at a different pixel position. This gives us:

$$\mathbf{b} = K\mathbf{i} + \mathbf{n}$$

Finally, we can write an equivalent representation of the imaging model with the Convolution Theorem as

$$\mathcal{F}\{b\} = \mathcal{F}\{i\}\mathcal{F}\{k\} + n$$

## 3.2 The Deconvolution Problem

The deconvolution problem, at its core, involves retrieving the ground truth image  $i$  from a convolved image  $b$ . This can be done in a *blind* setting, where the kernel  $k$  is unknown, or in a *non-blind* setting, where  $k$  is known. This project tackles the latter, where the PSF is known.

One key observation is that with a known PSF  $k$ , in a noise-free setting, frequency domain representation allows us to directly unfilter an image  $b$  to retrieve the ground truth image  $i$ :

$$\begin{aligned} b = i * k &\triangleq \mathcal{F}\{b\} = \mathcal{F}\{i\}\mathcal{F}\{k\} \\ \Rightarrow \mathcal{F}\{i\} &= \frac{\mathcal{F}\{b\}}{\mathcal{F}\{k\}} \\ \Rightarrow i &= \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{b\}}{\mathcal{F}\{k\}}\right\} \end{aligned}$$

However, such scenarios are unrealistic, and implementing the above equation in a real world setting with noise produces bad results (see Appendix A). To tackle this issue, image priors can be added to the problem in order to increase robustness. This is done in Wiener Deconvolution by taking into account signal-to-noise ratios (SNR) at different frequencies computed through natural image statistics.

In general, we express deconvolution as a constrained least-squares reconstruction problem, where the reconstructed image is regularized to have sparse gradients, common with natural images. More formally, we write the problem as

$$\hat{i} = \underset{x}{\operatorname{argmin}} \|Kx - b\|_2^2 + \lambda \|\nabla x\|_n^n$$

Where  $K$  is the matrix-vector representation of the convolution with the kernel  $k$ . In this project, we use the Total Variation (TV) constraint, which sets out to minimize the L1 norm of the image's gradient.

## 3.3 ADMM

The Alternating-Direction Method of Multipliers (ADMM) is a general framework for solving minimization problems in the form of

$$\begin{aligned} &\text{minimize } f(x) + g(z) \\ &\text{subject to } Ax + Bz = c \end{aligned}$$

The ADMM algorithm is used to solve problems in this form iteratively and efficiently by introducing a dual variable  $y$  that splits  $f(x)$  and  $g(z)$  into independent problem. This structure is usually exploited by formulating  $f$  and  $g$  such that their proximal operators are computationally cheap, allowing for low complexity operations within the iterative loop.

## 3.4 ADMM-based deconvolution

The deconvolution problem stated in Section 3.1 can be converted into an ADMM problem as:

$$\begin{aligned} f(x) &= \frac{1}{2} \|Kx - b\|_2^2 \\ g(z) &= \lambda \|z\|_1 \\ A &= \nabla, B = -I, c = 0 \end{aligned}$$

With this formulation, the objective can be minimized through the following algorithm:

---

**Algorithm 1** ADMM-TV for Deconvolution with constant PSF

---

```

1: Initialize:  $x^0 = 0, z^0 = 0, u^0 = 0$ 
2: for  $k = 0, 1, \dots, N - 1$  do
3:    $v^k \leftarrow z^k - u^k$ 
4:    $x^{k+1} \leftarrow \text{proximal}_x(v^k)$ 
5:    $v^k \leftarrow \nabla_x x^{k+1} + u^k$ 
6:    $z^{k+1} \leftarrow \text{proximal}_z(v^k)$ 
7:    $u^{k+1} \leftarrow u^k + \nabla_x z^{k+1} - z^{k+1}$ 
8: end for
9: return  $x^N$ 

```

---

The proximal operators for this formulation of the deconvolution problem have efficient closed-form solutions:

- $\text{proximal}_x(v)$  is an unfiltering operation:

$$x^{k+1} = \mathcal{F}^{-1} \left\{ \frac{\mathcal{F}\{K\}^* \mathcal{F}\{b\} + \rho(\mathcal{F}\{\nabla_x\}^* \mathcal{F}\{v_1\} + \mathcal{F}\{\nabla_y\}^* \mathcal{F}\{v_2\})}{\mathcal{F}\{K\}^* \mathcal{F}\{K\} + \rho(\mathcal{F}\{\nabla_x\}^* \mathcal{F}\{\nabla_x\} + \mathcal{F}\{\nabla_y\}^* \mathcal{F}\{\nabla_y\})} \right\}$$

Note that the only values computed per iteration are  $\mathcal{F}\{v_1\}$  and  $\mathcal{F}\{v_2\}$ , all other values can be pre-computed to increase the efficiency.

- $\text{proximal}_z(v)$  is the soft-thresholding operator with threshold  $\lambda/\rho$ :

$$z^{k+1} = \text{sign}(v) \max(|v| - \lambda/\rho, 0)$$

### 3.5 Douglas-Rachford Splitting

The Douglas-Rachford Splitting algorithm (DR) has a slightly different setup than ADMM. In it, we set to

$$\begin{aligned} & \text{minimize } f(x) + g(z) \\ & \text{subject to } Ax = z \end{aligned}$$

This is quite similar to the ADMM formulation. Like it, we use a dual variable  $y$  to split and connect  $x$  and  $z$ , and employ an iterative approach to the optimization.

### 3.6 Spatially-Varying PSF Deconvolution

We follow the Nagy-O’Leary [5] formulation for Spatially-Varying PSFs in the implemented deconvolution approach, mainly,

$$K = \sum_{p=1}^P U_p K_p$$

where  $K$  is the matrix representation of the spatial PSF kernel, with each  $K_p$  representing the constant PSF for a given patch of the image  $p$ , and each  $U_p$  being a diagonal matrix that weights how much different pixels are affected by different kernels. With this model, we can express the deconvolution problem as a DR problem as

$$\text{minimize } \frac{1}{2} \|Kx - b\|_2^2 + \lambda \|\nabla x\|_1$$

The key insight is that with the Nagy-O’Leary model, the fidelity term can be split into independent parts in the context of a DR problem, by letting

$$\begin{aligned} f(x) &= 0 \\ g(z_1, \dots, z_P, w) &= \frac{1}{2} \|U_1 z_1 + \dots + U_P z_P\|_2^2 + \lambda \|w\|_1 \\ A &= [K_1, \dots, K_P, \nabla] \end{aligned}$$

Where

$$Ax = [K_1 x, \dots, K_P x, \nabla x]$$

meaning that a solution that minimizes  $f(x) + g(z)$  where  $Ax = z$  minimizes

$$0 + g(Ax) = \frac{1}{2} \|U_1 K_1 x + \dots + U_P K_P x\|_2^2 = \frac{1}{2} \|Kx - b\|_2^2 + \lambda \|\nabla x\|_1$$

The iterative solver is closely adapted from the original author’s of [6] MATLAB implementation<sup>2</sup>

## 4 METHODS

The main challenges of this project were to implement the algorithm outlined in section 5.4 of [4] and to accurately record the spatially varying PSF of my phone’s damaged lens.

### 4.1 Algorithm implementation

All of the code used in this project can be found in the following open sourced repository<sup>3</sup>. I implemented the discussed algorithms in Python making use of SciPy<sup>4</sup>, NumPy<sup>5</sup> and OpenCV<sup>6</sup>. Usage and more implementation detail can be found at the repository.

As previously mentioned, the Python implementation of the Douglas-Rachford TV deconvolution is heavily based on O’Connor and Vandenberghe’s MATLAB implementation, including greater detail not explicitly described in paper. For a more involved discussion of the implementation, see Appendix C.2.

### 4.2 PSF estimation

Using a linear stage, I translated a point light source relative to the camera’s optical center and recorded its PSF directly on smaller patches of the image. This was especially complicated as the camera in question had a wide angle field of view of 120 degrees, making it difficult to translate the light source and camera without losing light. To account for this, I normalized all PSFs equally. The resulting PSF measurement can be seen in Figure 2.

## 5 RESULTS

Comparison of runtime and quantitative quality metrics for single PSF vs. spatially-varying PSF. Qualitative comparison of methods

In order to evaluate the presented algorithms, synthetic data was manufactured to model the constant PSF case and the spatially-varying PSF case.

<sup>2</sup>[https://github.com/danielvoconnor/TV\\_deblur\\_spaceVariantKernel/blob/master/tvDeblur\\_varBlur\\_freeBCs\\_DR.m](https://github.com/danielvoconnor/TV_deblur_spaceVariantKernel/blob/master/tvDeblur_varBlur_freeBCs_DR.m)

<sup>3</sup><https://github.com/fzmautner/463-final-project>

<sup>4</sup><https://scipy.org/>

<sup>5</sup><https://numpy.org/>

<sup>6</sup><https://opencv.org/>

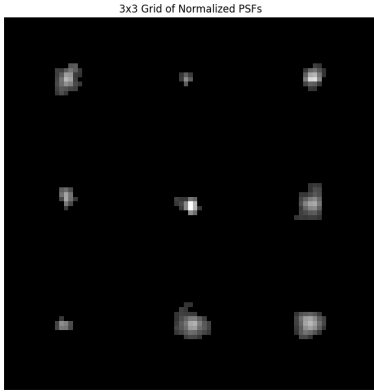


Figure 2:  $3 \times 3$  grid of Spatially-Varying PSF of my damaged phone’s lens

### 5.1 Spatially-Invariant PSF

This section evaluates the performance of the ADMM-TV algorithm and compares it to a baseline Wiener Deconvolution approach. The ADMM-TV algorithm shows significant improvement over Wiener deconvolution in both noise handling and edge preservation, as shown in Figure 3.

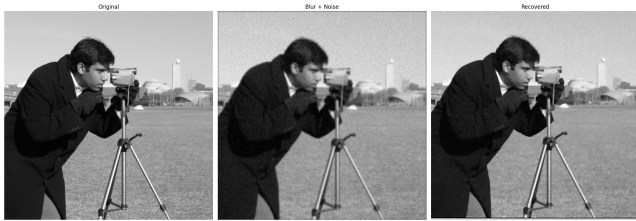


Figure 3: Left: Original image. Middle: Image with gaussian blur and noise. Right: Image recovered with ADMM-TV deconvolution after 30 iterations.

A quick comparison to an off-the-shelf Wiener deconvolution method by scikit-image (with the same PSF) reveals, as expected, better qualitative performance by ADMM-TV, avoiding the artifacts seen in the Wiener deconvolved image while maintaining detail restoration.

The regularization parameter  $\lambda$  plays a crucial role in the quality of reconstruction. Higher values of  $\lambda$  result in smoother images with less noise but at the cost of losing fine detail, while lower values preserve more detail but may amplify noise. Figure 5 demonstrates this tradeoff across different  $\lambda$  values.

ADMM-TV’s efficient runtime allows for channel-wise deblurring of color images with minimal computational overhead. The runtime of ADMM-TV is discussed alongside that of DR-TV in Section 5.2. Figure 6 shows the algorithm’s effectiveness on color images.

However, the assumption that the same PSF is applied over each kernel fails to consider the effects of chromatic aberration that a real lens can cause, especially if simple or damaged. This is discussed in more detail in Section 6.1 under Color and Chromatic Aberration.

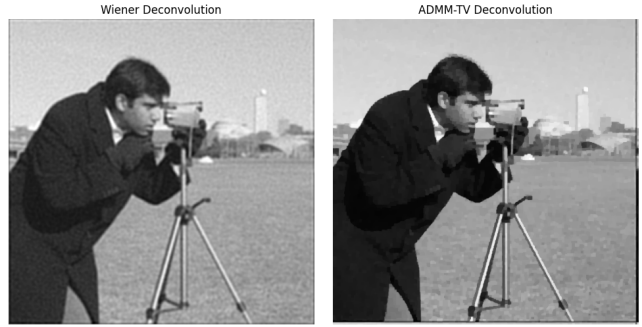


Figure 4: Left: Image recovered with ADMM-TV deconvolution after 30 iterations. Right: Image recovered with skimage.restoration.wiener with the best balance parameter.

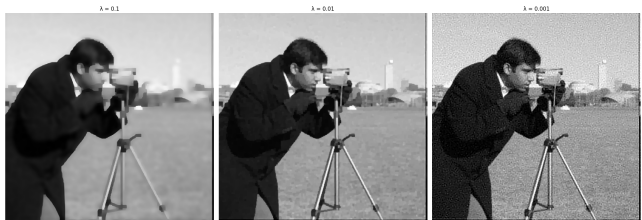


Figure 5: Effect of different  $\lambda$  values on reconstruction quality. Left:  $\lambda = 0.1$  showing over-smoothing. Middle:  $\lambda = 0.01$  balancing detail and noise. Right:  $\lambda = 0.001$  preserving detail but amplifying noise.

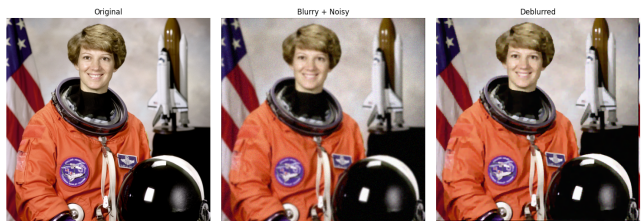
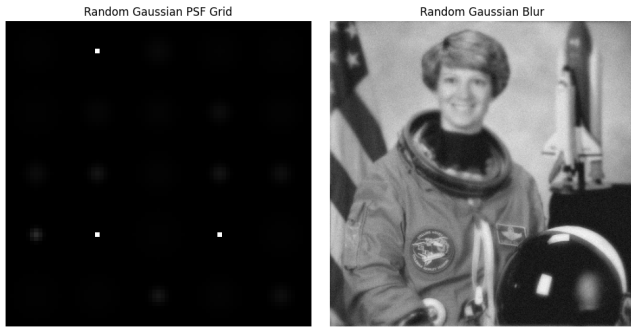


Figure 6: Color image deconvolution. Left: Original color image. Middle: Blurred and noisy image. Right: ADMM-TV reconstruction.

### 5.2 Spatially-Varying PSF

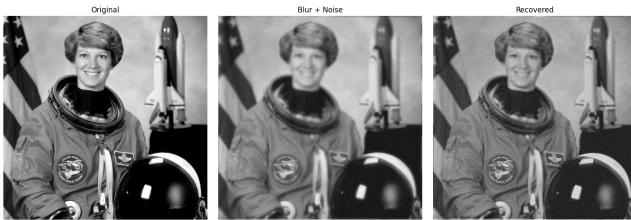
This section evaluates the performance of the DR-TV algorithm and compares it to assuming an invariant PSF deconvolved with ADMM-TV.

To simulate a damaged mobile phone lens, a spatially-varying PSF is designed to have local gaussian PSFs with randomly distributed standard deviations. These local PSFs are weighted at each pixel according to its distance from the center of the patch (See Appendix C.1). Notice in Figure 7 that the astronaut’s badge remains relatively sharp while her face is heavily blurred, reflecting a wider local PSF kernel at that patch in the image.



**Figure 7: Left: Spatially-Varying PSF with randomly distributed gaussian local PSFs. Right: Effect of varied PSF over image showing non-uniform blurring effect.**

The DR-TV algorithm is capable of retrieving significant detail in the more blurred parts of the image, and the overall result looks more consistent. However, some detail is lost, and much of the work lies in the hand of the end user's choice of hyperparameters, which can significantly affect the final result.



**Figure 8: Left: Original unblurred and image. Middle: Original image with non-local blur and gaussian noise added to it. Right: Recovered image with DR-TV after 100 iterations.**

In a scenario such as this, the best that ADMM-TV can do is assume a single, average PSF kernel, and treat it uniformly over the image:



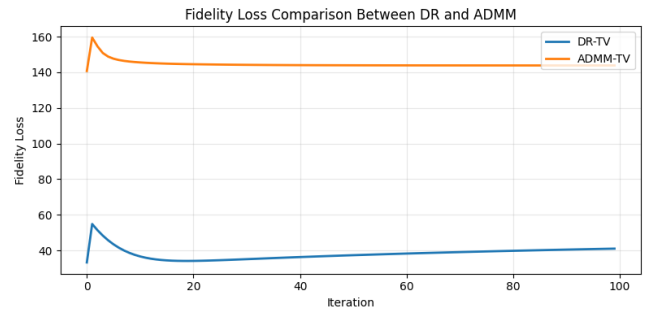
**Figure 9: Left: Image recovered with DR-TV. Right: Image recovered with ADMM-TV.**

From using a single approximate kernel for the entire image, the ADMM-TV solver is unable to reconstruct the image with spatial

awareness, and the end result looks less precise than its DR-TV counterpart at similar noise levels.

However, the two algorithms vary greatly in terms of computational cost. Both were run for 100 iterations on an Apple M1 chip. The DR-TV algorithm ran for 4 minutes and 14 seconds (2.54 iterations per second), compared to just four seconds (21.51 iterations per second) for the ADMM-TV implementation. This reflects in a speedup of over 8x, which while not insignificant, is much lower than 25x, which would reflect a naïve implementation of deconvolution with now 25 unique PSF kernels.

Finally, the loss curves for each method confirm the dominance of DR-TV in this scenario: Noticeably, both methods converged at

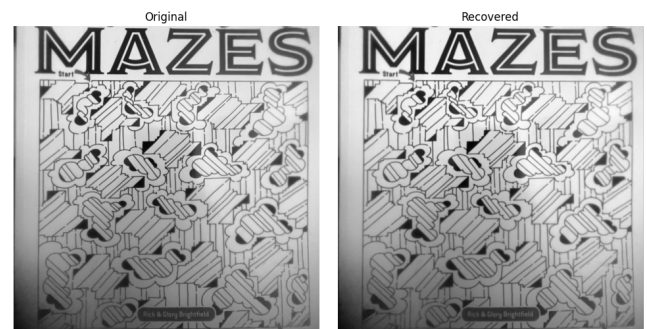


**Figure 10: Fidelity loss over iterations for DR-TV (blue) and ADMM-TV (orange).**

approximately 20 iterations, but whereas ADMM-TV plateaued, the DR-TV solution went on to slowly degrade.

### 5.3 Real World Spatially-Varying PSF

Using the same pipeline outline above, I used my phone's calibrated Spatially-Varying PSF to perform deconvolution:



**Figure 11: Left: Original Image captured from damaged phone lens. Right: Deconvolved image with measured Spatial PSF.**

At a first glance, the effects of deconvolution aren't very noticeable. This is likely due to an inaccurate PSF measurement or suboptimal hyperparameters. Taking a closer look at the corners, where the blur is strongest, however, indicates an improved image with less noise:

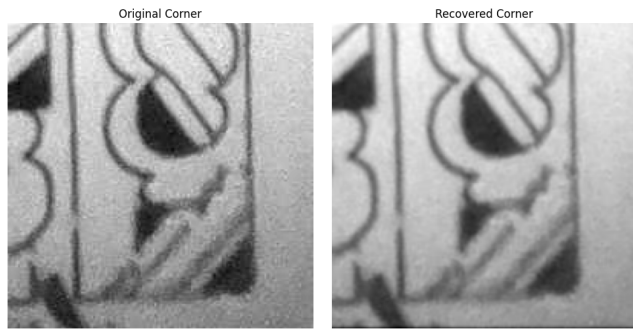


Figure 12: Left: Zoom in in Original Image. Right: Zoom in in Deconvolved image.

## 6 LIMITATIONS AND FUTURE WORK

While the methods presented in this work show promising results for deblurring images with various PSFs, there are several limitations and opportunities for future work:

### 6.1 Color and Chromatic Aberration

The current implementation processes works on single channel images, and color images can be composed by independently deconvoluting each channel. The implicit assumption that the PSF is constant under different channels does not reflect real world conditions, where the wavelength of incoming light plays an important role in the lens' PSF. This is particularly limiting when dealing with chromatic aberration, which is especially common in simple, cheap lenses. For a more complete treatment of color in spatially-varying deblurring, see [4].

### 6.2 Future Directions

Several promising directions could extend this work:

- **Machine Learning Integration:** Recent work has shown success incorporating deep neural networks into traditional optimization frameworks. (CITE some).
- **Automated PSF Estimation:** The current approach requires manual PSF measurement. An automated method to estimate the spatially-varying PSF directly from blurred images would make the technique more practical.
- **Blind Deconvolution:** The current approach relies on the user knowing the PSF of their lens. This may not always be practical, especially in commercial applications such as damaged mobile camera imaging enhancement. As such, blind deconvolution approaches such as (CITE 3 or so) may be more appropriate.
- **Real-time Processing:** The iterative nature of the algorithm makes it computationally intensive. Exploring parallel implementations or more efficient optimization techniques could enable real-time applications.

## ACKNOWLEDGMENTS

I would like to thank Professor Gkioulekas and Dorian Chan for their teaching and guidance throughout the semester. I would also

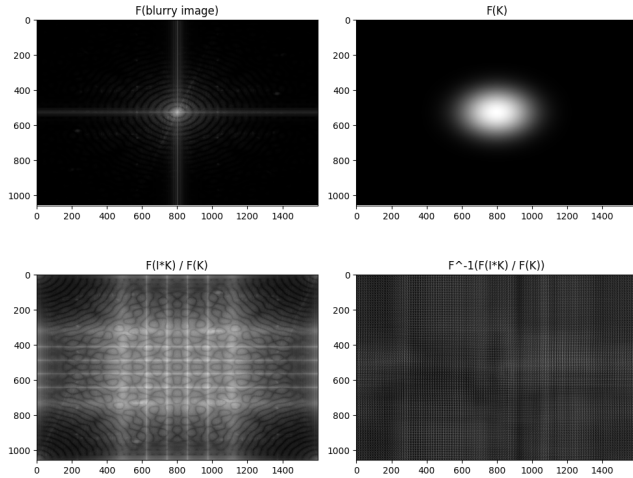
like to thank O'Connor and Vandenberghe for open-sourcing their DR-TV implementation.

## REFERENCES

- [1] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning* 3 (01 2011), 1–122. <https://doi.org/10.1561/22000000016>
- [2] A. Chambolle and Thomas Pock. 2011. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision* 40 (2011), 120–145. <https://api.semanticscholar.org/CorpusID:261281173>
- [3] R. Glowinski and A. Marroco. 1975. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique* 9, R2 (1975), 41–76. [http://www.numdam.org/item/M2AN\\_1975\\_\\_9\\_2\\_41\\_0/](http://www.numdam.org/item/M2AN_1975__9_2_41_0/)
- [4] Felix Heide, Mushfiqur Rouf, Matthias B. Hullin, Bjorn Labitzke, Wolfgang Heidrich, and Andreas Kolb. 2013. High-quality computational imaging through simple lenses. *ACM Trans. Graph.* 32, 5, Article 149 (Oct. 2013), 14 pages. <https://doi.org/10.1145/2516971.2516974>
- [5] James G. Nagy and Dianne P. O'Leary. 1998. Restoring Images Degraded by Spatially Variant Blur. *SIAM Journal on Scientific Computing* 19, 4 (1998), 1063–1082. <https://doi.org/10.1137/S106482759528507X> arXiv:<https://doi.org/10.1137/S106482759528507X>
- [6] Daniel O'Connor and Lieven Vandenberghe. 2017. Total variation image deblurring with space-varying kernel. *Computational Optimization and Applications* 67 (07 2017). <https://doi.org/10.1007/s10589-017-9901-1>
- [7] William Hadley Richardson. 1972. Bayesian-Based Iterative Method of Image Restoration\*. *J. Opt. Soc. Am.* 62, 1 (Jan 1972), 55–59. <https://doi.org/10.1364/JOSA.62.000055>

## A NAÏVE UNFILTERING AND WIENER DECONVOLUTION

Even with minimal added noise and blur, Naïve deconvolution by unfiltering rarely works, as illustrated below:

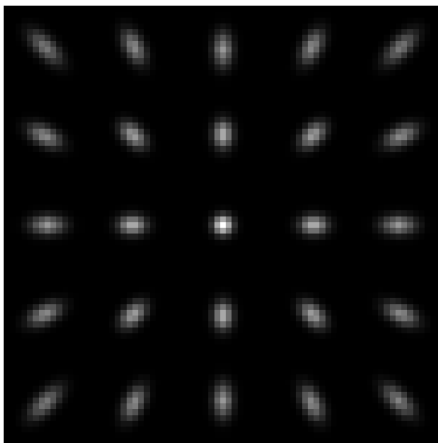


**Figure 13: Naïve unfiltering results. Top Left: Fourier Transform of original blurry image. Top Right: Fourier Transform of PSF. Bottom Left: Fourier transform of unfiltering. Bottom Right: Inverse Fourier Transform of unfiltered image.**

## B MORE RESULTS

### B.1 DR-TV

One interesting Space-Varying PSF to model is that of a simple lens that suffers from geometric distortion. This type of aberration manifests as objects near the edges of the image appearing stretched out. We can simulate this effect using a Non-Local PSF by simply stretching out the local gaussian kernels proportionally to their distance from the center of the PSF grid:

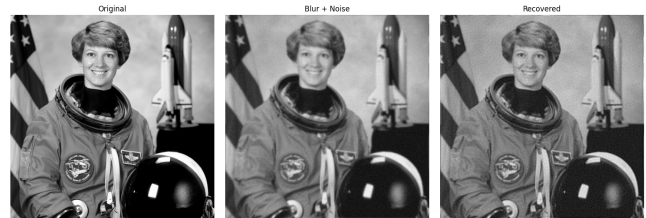


**Figure 14: Radially distorted PSF grid**



**Figure 15: Radially distorted and blurred image**

Applying this to the image:  
DR-TV is able to successfully recover the original image:



**Figure 16: Left: Original. Middle: Radially distorted and noisy image. Right: Recovered**



**Figure 17: Isolated reconstructed image.**

## C IMPLEMENTATION DETAILS

### C.1 PSF Modelling

Unlike [6], which uses a mostly rectangular weighing function for image patches, the weights used in this project follow a natural decay established as a function of the distance of a pixel to the center of the local PSF's patch. This is illustrated below:

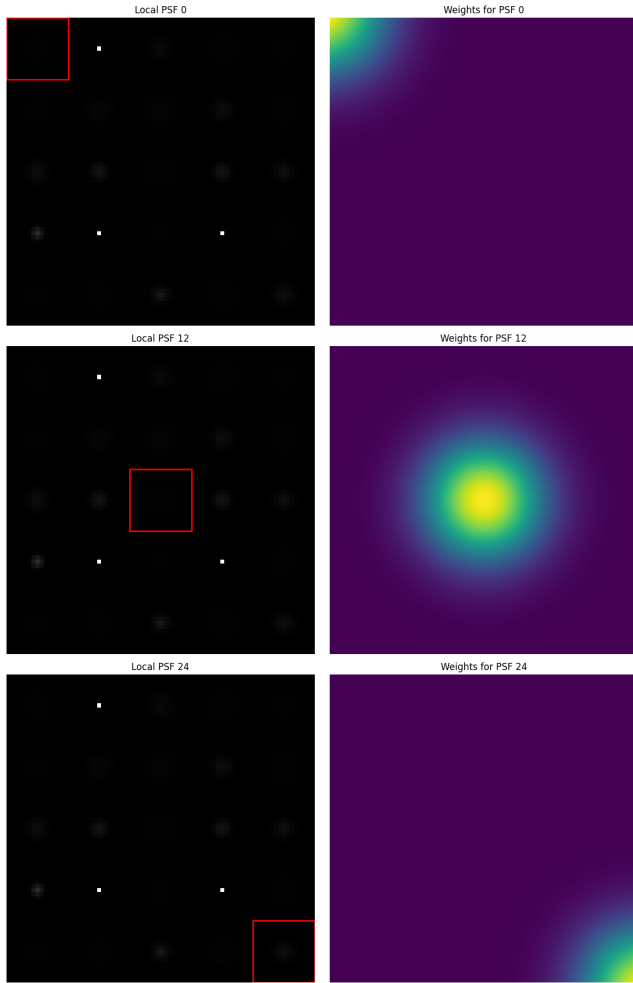


Figure 18: Weight matrices  $U$  for different local PSFs

### C.2 Algorithm Implementation

The Douglas-Rachford splitting algorithm was implemented in Python following O'Connor and Vandenberghe's original MATLAB code. The algorithm requires careful handling of boundary conditions, using circular boundary conditions for the convolution operations and thus requiring zero-padding to the image during the deconvolution process. This is handled internally and the padding size is computed as half of the a local PSF's size.