

# Comparison of Image Recovery Algorithms

Final Project - COMPSCI 590OP - Applied Numerical Optimization

Kartik Choudhary  
University of Massachusetts Amherst

December 18, 2022

## 1 Introduction

Digital images often get corrupted due to various factors - noise picked up by the camera sensor while capturing the image, distortions by the image processing algorithm, losses and other artifacts due to compression, artifacts due to encoding/decoding images into different formats, etc.

These distortions are undesirable and we want to recover the original image  $\mathbf{x}$  whose distorted version  $\mathbf{y}$  is available to us, along with some information/assumptions about the process that corrupted the image. The goal is to retrieve an estimate  $\hat{\mathbf{x}}$ , that is as close as possible to the original image  $\mathbf{x}$ , based on some distance metric that can quantify the closeness of two images. There are many classical signal processing algorithms (Banham and Katsaggelos, 1997) as well as newer approaches like LCG (Ehlers and Stuckey, 2016) that pose the reconstruction of the images as an optimization problem.

In this report we will focus on two types of distortion - additive white Gaussian noise (AWGN) and Gaussian blur - and compare two different denoising algorithms - Plug-and-Play ADMM and Total Variation denoising - for how well they are able to recover the original image under different levels of distortion.

## 2 Problem Setup

As described by Tirer and Giryes (2017), the problem of image restoration can be formulated by the following equation

$$\mathbf{y} = H\mathbf{x} + \mathbf{e} \quad (1)$$

Where  $\mathbf{y} \in \mathbb{R}^m$  is the observed image, which is obtained by applying a distortion operation  $H \in \mathbb{R}^{m \times n}$  on the original image  $\mathbf{x} \in \mathbb{R}^n$  and then adding an additive noise  $\mathbf{e} \in \mathbb{R}^m$ .

In this report, we will consider the case when  $H$  is a Gaussian blurring operator (Segall and Katsaggelos, 2022), and  $\mathbf{e}$  is additive white Gaussian noise (AWGN) (Liu and Lin, 2013).

To reverse the effect of the degradation operations in Equation 1, we want to first remove the AWGN from the observed image, and then de-blur it to recover the original image.

Following the approach of Chan et al. (2016), we want the recovered image  $\hat{\mathbf{x}}$  to be such that it maximizes the posterior probability of the original image  $\mathbf{x}$  given the observation  $\mathbf{y}$

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}) \quad (2)$$

Equivalently, we can also minimize the negative log of the posterior

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} -\log p(\mathbf{y}|\mathbf{x}) - \log p(\mathbf{x}) \quad (3)$$

Where  $p(\mathbf{y}|\mathbf{x})$  describes the image distortion process, and the  $p(\mathbf{x})$  is the prior distribution of the original image.

This can be modeled as an unconstrained optimization problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} f(\mathbf{x}) + \lambda g(\mathbf{x}) \quad (4)$$

Where

$$f(\mathbf{x}) := -\log p(\mathbf{y}|\mathbf{x})$$

$$g(\mathbf{x}) := -\frac{1}{\lambda} \log p(\mathbf{x})$$

### 3 Methodology

We will be implementing the Plug-and-Play ADMM algorithm to solve the optimization problem described in Section 2 based on descriptions by Chan et al. (2016) and Fan et al. (2019). The complete algorithm and derivation is given in Section 4.

We will also compare the results of this algorithm with the anisotropic total variation (TV) denoising algorithm (Weickert, 1998). This algorithm is an extension of the 1-D TV denoising done in Assignment 5, and will be based on the implementation in the Kornia library (Riba et al., 2020; Shi et al., 2020) using the PyTorch autograd package (Paszke et al., 2017).

We will compare the performance of the two algorithms on the  $512 \times 512$  image shown in Figure 1 under the following conditions (more details in Section 5)-

1. Effect on original image - We want the algorithms to introduce minimal artifacts in images when there are no distortions
2. Performance in presence of low additive noise and no blur
3. Performance in presence of no additive noise and low amount of blur
4. Performance in presence of low additive noise and low amount of blur
5. Performance in presence of high additive noise and high amount of blur



Figure 1: Different types of noisy images that will be given as input to our algorithms

## 4 Algorithms

### 4.1 Plug-and-Play ADMM

In the ADMM algorithm we convert the unconstrained optimization into a constrained optimization problem

$$\hat{\mathbf{x}}, \hat{\mathbf{z}} = \arg \min_{\mathbf{x}, \mathbf{z}} f(\mathbf{x}) + \lambda g(\mathbf{z}); \quad \text{subject to } \mathbf{x} = \mathbf{z} \quad (5)$$

Now, we can write the augmented Lagrangian as

$$\mathcal{L}(\mathbf{x}, \mathbf{z}, \boldsymbol{\gamma}) = f(\mathbf{x}) + \lambda g(\mathbf{z}) + \boldsymbol{\gamma}^\top (\mathbf{x} - \mathbf{z}) + \frac{t}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 \quad (6)$$

To solve this, we iteratively apply the following steps to the scaled version of the problem

$$\mathbf{x}_{k+1} = \arg \min_{\mathbf{x}} f(\mathbf{x}) + \frac{t}{2} \|\mathbf{x} - (\mathbf{z}_k - \boldsymbol{\gamma}_k)\|_2^2 \quad (7)$$

$$\mathbf{z}_{k+1} = \arg \min_{\mathbf{z}} \lambda g(\mathbf{z}) + \frac{t}{2} \|\mathbf{z} - (\mathbf{x}_{k+1} + \boldsymbol{\gamma}_k)\|_2^2 \quad (8)$$

$$\boldsymbol{\gamma}_{k+1} = \boldsymbol{\gamma}_k + (\mathbf{x}_{k+1} - \mathbf{z}_{k+1}) \quad (9)$$

Now, in the Plug-and-Play (PnP) algorithm (Venkatakrishnan et al., 2013), we do not define the prior ourselves, but instead, use an off-the-shelf denoiser  $\mathcal{D}_\sigma$  as an approximation to the  $\mathbf{z}$  update step, where  $\sigma$  is the parameter determining the strength of the denoiser. The denoiser used in our algorithm is the BM3D denoiser (Dabov et al., 2007).

We will also increase  $t_{k+1} = \kappa t_k$  for some parameter  $\kappa > 1$ . The parameter controlling the strength of the denoiser in the  $k^{th}$  iteration is given as  $\sigma_k = \sqrt{\frac{\lambda}{t_k}}$ .

Combining all these steps, the final algorithm for Plug-and-Play ADMM image restoration algorithm becomes

---

**Algorithm 1** Pseudocode for Plug-and-Play ADMM [Adapted from Chan et al. (2016)]

---

```

1: Input:  $t_0, \lambda, \kappa > 1$ 
2: Initialize  $\mathbf{x}_0, \mathbf{z}_0, \boldsymbol{\gamma}_0$ 
3:  $k = 0$ 
4: while not converged do
5:    $\mathbf{x}_{k+1} = \arg \min_{\mathbf{x}} f(\mathbf{x}) + \frac{t}{2} \|\mathbf{x} - (\mathbf{z}_k - \boldsymbol{\gamma}_k)\|_2^2$ 
6:    $\sigma_k = \sqrt{\lambda/t_k}$ 
7:    $\mathbf{z}_{k+1} = \mathcal{D}_{\sigma_k}(\mathbf{x}_{k+1} + \boldsymbol{\gamma}_k)$ 
8:    $\boldsymbol{\gamma}_{k+1} = \boldsymbol{\gamma}_k + (\mathbf{x}_{k+1} - \mathbf{z}_{k+1})$ 
9:    $t_{k+1} = \kappa t_k$ 
10:   $k \leftarrow k + 1$ 
11: end while

```

---

The  $\mathbf{x}$  update can be done using gradient descent. The parameter  $\lambda$  is the regularization parameter for this algorithm which determines the strength of the denoising.

The stopping criteria is defined as when the change in the three parameters being updated becomes small. We define  $\Delta_{k+1}$  as

$$\Delta_{k+1} = \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 + \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2^2 + \|\boldsymbol{\gamma}_{k+1} - \boldsymbol{\gamma}_k\|_2^2 \quad (10)$$

We consider the algorithm to have converged when  $\Delta_{k+1} < tol$  for some tolerance  $tol$  that is defined before running the algorithm.

## 4.2 Total Variation Denoising

For comparison with our implementation, we will use Total Variation (TV) denoising algorithm designed to remove Gaussian blur and AWGN (see Rodríguez, 2013).

For the TV denoising algorithm, we are trying to solve the following optimization problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\| + \lambda |\nabla \mathbf{x}| \quad (11)$$

Where  $|\nabla \mathbf{x}|$  is the total variation of the image and  $\lambda$  is the hyperparameter regulating the strength of regularization.

For 2-D images, there are two ways of defining the total variation.

The classic isotropic version is given by

$$|\nabla \mathbf{x}_{i,j}| = \sqrt{(\mathbf{x}_{i+1,j} - \mathbf{x}_{i,j})^2 + (\mathbf{x}_{i,j+1} - \mathbf{x}_{i,j})^2} \quad (12)$$

This results in the final image being blurry and does not preserve sharp edges very well. An alternative is to use the anisotropic total variation that is given by

$$|\nabla \mathbf{x}_{i,j}| = |\mathbf{x}_{i+1,j} - \mathbf{x}_{i,j}| + |\mathbf{x}_{i,j+1} - \mathbf{x}_{i,j}| \quad (13)$$

This version better preserves sharp edges and creates images that look piecewise constant, and more cartoon-like (see González et al., 2017 for more details). We will be using this version in our implementation.

In Assignment 5 we have already implemented 1-D total variation using efficient implementations of matrices  $D$ ,  $D^\top$ , and  $DD^\top$ , where  $D\mathbf{x}$  is the 1-D total variation of the vector  $\mathbf{x}$ .

Extending this definition to 2-D, we can define the total variation gradient operator  $\tilde{D}$  as

$$\tilde{D} = [D_h, D_v] \quad (14)$$

Where  $D_h$  and  $D_v$  are the horizontal and vertical 1-D partial derivative operators similar to the ones we have implemented in the assignment. Thus, the total variation (anisotropic) of an image  $\mathbf{x}$  can be written as

$$|\nabla \mathbf{x}| = \|\tilde{D}\mathbf{x}\|_1 = \|D_h\mathbf{x}\|_1 + \|D_v\mathbf{x}\|_1 \quad (15)$$

Now, this problem can be solved in the dual domain like the assignment using proximal gradient where the proximal operator just clips the input to  $[-\lambda, \lambda]$ .

## 5 Results

We now show the results of the two algorithms under different levels of noise (see Appendix A for additional results). We have introduced two types of distortions, and in different test they are present in low or high amounts, which is defined as-

1. Blur: Standard deviation of the Gaussian kernel - 1.0 for low blur and 2.0 for high blur.
2. AWGN: Standard deviation of i.i.d. additive Gaussian noise with zero mean - 0.1 for low noise and 0.5 for high noise.

For the PnP ADMM algorithm, as seen in Figure 2, we observe that it can remove additive noise reasonably well under mildly noisy conditions, and the resulting image is looks reasonably similar to the original undistorted image. However, it is not very effective in removing Gaussian blur, which stays in the image even if the amount of blur in the input was low, and in fact the algorithm introduces some blurring of its own if the regularization parameter is too high. We also note that the algorithm is not able to recover the image well if the input image is very highly distorted.

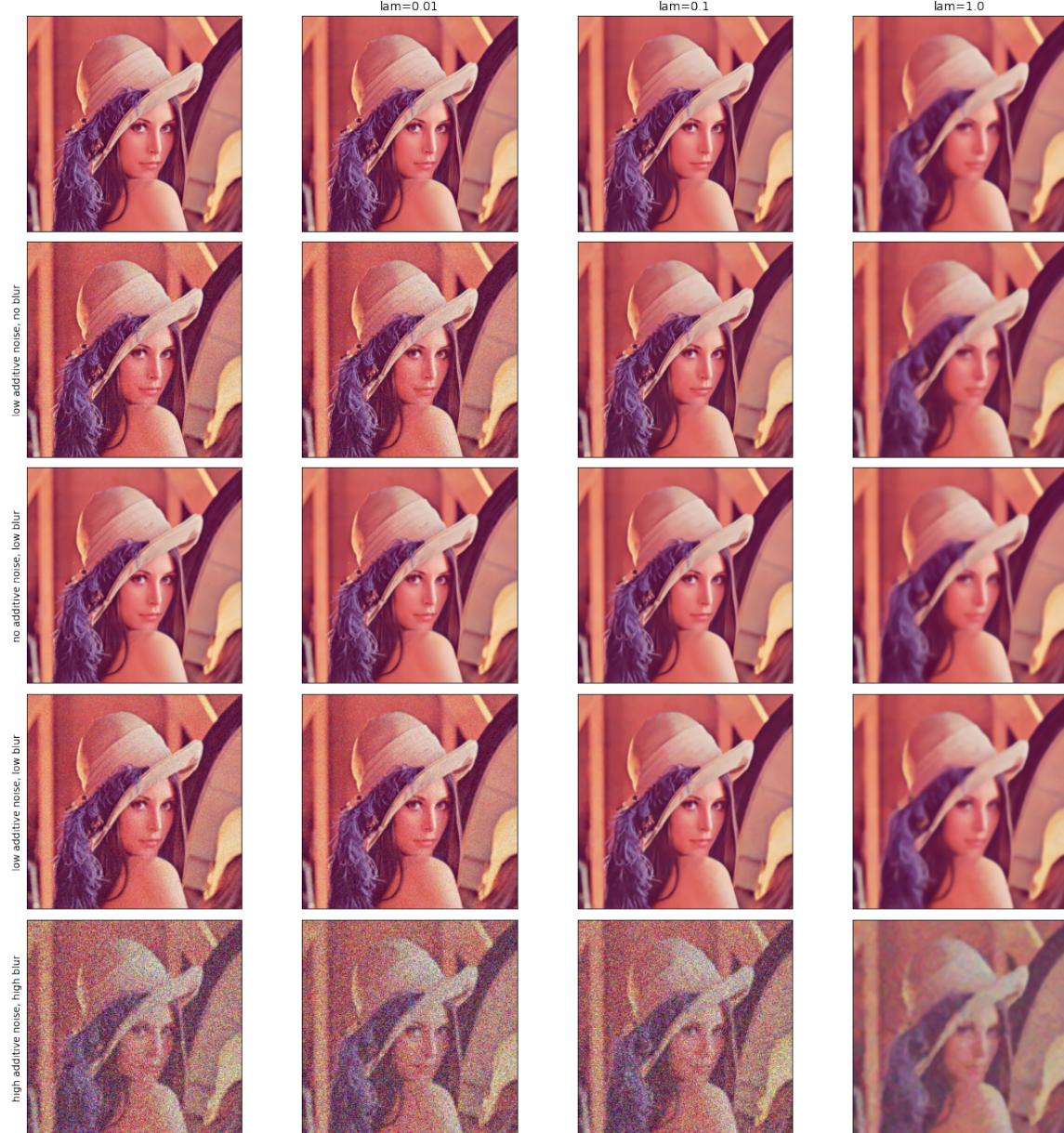


Figure 2: Results of the PnP ADMM algorithm for different levels of regularization under different levels of noise (Original image on the top left corner)

For the Total Variation algorithm, as seen in Figure 3, we notice slightly different characteristics. We observe that the algorithm is better at counteracting Gaussian blur, while its capabilities for removing additive noise are not as strong compared to PnP ADMM. Increasing the regularization parameters to improve additive noise removal strength introduces the characteristic “cartoonizing” artifacts in the result, and the algorithm introduces sharp edges even in places where it is not present in the original image (such as the out-of-focus background). Again, the algorithm is not able to recover the image well under high levels of distortion.

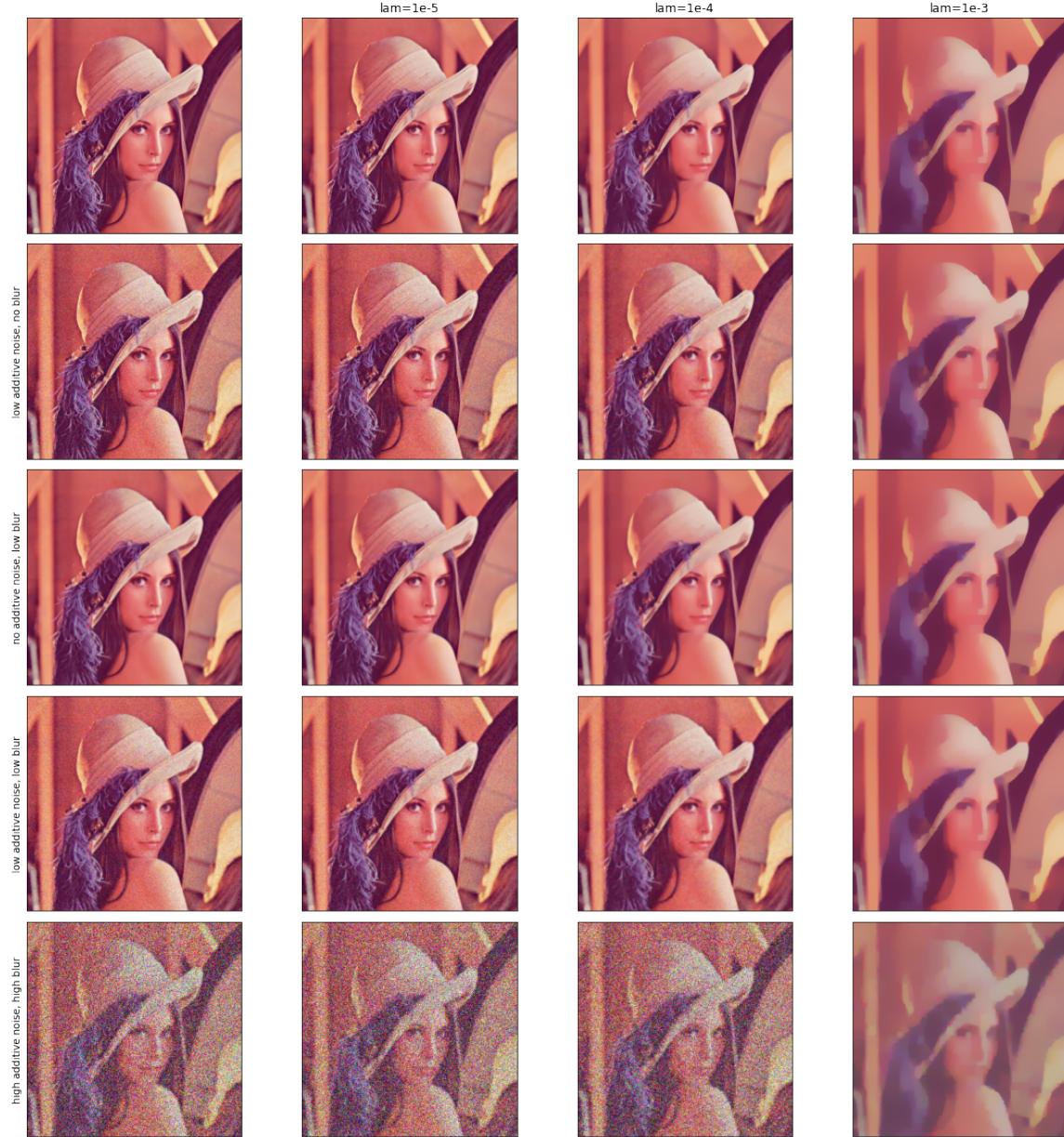


Figure 3: Results of the Total Variation denoising algorithm for different levels of regularization under different levels of noise (Original image on the top left corner)

## 6 Future Work

In our implementation of the PnP ADMM algorithm we have chosen to update the parameter  $t$  by a constant factor in each iteration. We can also use the adaptive update rule based on Goldstein et al. (2014), which uses the relative residue defined as

$$\Delta_{k+1} = \frac{1}{\sqrt{n}} \left( \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 + \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2^2 + \|\boldsymbol{\gamma}_{k+1} - \boldsymbol{\gamma}_k\|_2^2 \right) \quad (16)$$

Where  $n$  is the size of the image. Now, for any  $\eta \in [0, 1)$ , and a constant  $\kappa > 1$ , we conditionally update  $t$  such that

$$t_{k+1} = \begin{cases} \kappa t_k & \Delta_{k+1} \geq \eta \Delta_k \\ t_k & \Delta_{k+1} < \eta \Delta_k \end{cases} \quad (17)$$

For the Total Variation algorithm, we have used anisotropic total variation as our regularization term. To get a better idea of how this regularization term affects the final image compared to isotropic total variation, we can also implement the isotropic version of total variation, or a weighted difference of the two (Lou et al., 2015).

In this report we have looked at distortions caused by AWGN and Gaussian blurring. In the real world, there are many other types of distortions that can be present in an image, and we do not know in advance which types of distortions and with what strength are present in the image. The PnP ADMM algorithm implemented by us can be made more robust by implementing additional steps that can deal with more types of distortions.

## Acknowledgement

We would like to thank Prof. Andrew Lan for teaching the course COMPSCI 590OP - Applied Numerical Optimization, and for their guidance throughout the course and during this project. The course has given us in-depth knowledge of many fundamentals of optimization techniques and shown the importance of setting parameters like the step-size in gradient-based algorithms and the regularization parameters, that are sometimes set as an afterthought.

We are also thankful to the TAs, Yunfei Luo and Xi Chen for always being available to solve any issues or doubts regarding the assignments or theory via the Piazza forum.

## References

- Banham, M.R. and A.K. Katsaggelos (1997). “Digital image restoration.” In: *IEEE Signal Processing Magazine* 14.2, pp. 24–41. DOI: 10.1109/79.581363.
- Weickert, Joachim (1998). *Anisotropic diffusion in image processing*. English. Eur. Consortium Math. Ind. Stuttgart: Teubner. ISBN: 3-519-02606-6.
- Dabov, Kostadin et al. (2007). “Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering.” In: *IEEE Transactions on Image Processing* 16.8, pp. 2080–2095. DOI: 10.1109/TIP.2007.901238.
- Liu, Wei and Weisi Lin (2013). “Additive White Gaussian Noise Level Estimation in SVD Domain for Images.” In: *IEEE Transactions on Image Processing* 22.3, pp. 872–883. DOI: 10.1109/TIP.2012.2219544.
- Rodríguez, Paul (2013). “Total Variation Regularization Algorithms for Images Corrupted with Different Noise Models: A Review.” In: *Journal of Electrical and Computer Engineering* 2013, p. 217021. ISSN: 2090-0147. DOI: 10.1155/2013/217021. URL: <https://doi.org/10.1155/2013/217021>.
- Venkatakrishnan, Singanallur V. et al. (2013). “Plug-and-Play priors for model based reconstruction.” In: *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948. DOI: 10.1109/GlobalSIP.2013.6737048.
- Goldstein, Tom et al. (2014). “Fast Alternating Direction Optimization Methods.” In: *SIAM Journal on Imaging Sciences* 7.3, pp. 1588–1623. DOI: 10.1137/120896219. eprint: <https://doi.org/10.1137/120896219>. URL: <https://doi.org/10.1137/120896219>.
- Lou, Yifei et al. (2015). “A Weighted Difference of Anisotropic and Isotropic Total Variation Model for Image Processing.” In: *SIAM Journal on Imaging Sciences* 8.3, pp. 1798–1823. DOI: 10.1137/14098435X. eprint: <https://doi.org/10.1137/14098435X>. URL: <https://doi.org/10.1137/14098435X>.
- Chan, Stanley H. et al. (2016). “Plug-and-Play ADMM for Image Restoration: Fixed Point Convergence and Applications.” In: *CoRR* abs/1605.01710. arXiv: 1605.01710. URL: <http://arxiv.org/abs/1605.01710>.
- Ehlers, Thorsten and Peter J. Stuckey (2016). “Parallelizing Constraint Programming with Learning.” In: *Integration of AI and OR Techniques in Constraint Programming*. Ed. by Claude-Guy Quimper. Cham: Springer International Publishing, pp. 142–158. ISBN: 978-3-319-33954-2.
- González, Gerardo et al. (2017). “Isotropic and anisotropic total variation regularization in electrical impedance tomography.” In: *Computers and Mathematics with Applications* 74.3, pp. 564–576. ISSN: 0898-1221. DOI: <https://doi.org/10.1016/j.camwa.2017.05.004>. URL: <https://www.sciencedirect.com/science/article/pii/S0898122117302833>.
- Paszke, Adam et al. (2017). “Automatic differentiation in PyTorch.” In.
- Tirer, Tom and Raja Giryes (2017). “Image Restoration by Iterative Denoising and Backward Projections.” In: *CoRR* abs/1710.06647. arXiv: 1710.06647. URL: <http://arxiv.org/abs/1710.06647>.
- Fan, Linwei et al. (2019). “Brief review of image denoising techniques.” In: *Visual Computing for Industry, Biomedicine, and Art* 2.1, p. 7. ISSN: 2524-4442. DOI: 10.1186/s42492-019-0016-7. URL: <https://doi.org/10.1186/s42492-019-0016-7>.
- Riba, E. et al. (2020). “Kornia: an Open Source Differentiable Computer Vision Library for PyTorch.” In: *Winter Conference on Applications of Computer Vision*. URL: <https://arxiv.org/pdf/1910.02190.pdf>.
- Shi, Jian et al. (2020). *Differentiable Data Augmentation with Kornia*. arXiv: 2011.09832 [cs.CV].
- Segall, C. and Aggelos Katsaggelos (Dec. 2022). “Digital Image Restoration—Classical.” In.

## A Additional results

Shown below are the results for the same experiment as Section 5 on a  $512 \times 512$  grayscale image for the PnP ADMM algorithm.



Figure 4: Results of the Pnp ADMM algorithm for different levels of regularization under different levels of noise (Original image on the top left corner)

Results shown below are for the Total Variation denoising algorithm.



Figure 5: Results of the Total Variation denoising algorithm for different levels of regularization under different levels of noise (Original image on the top left corner)