

# Plug and Play ADMM

Mirza Fuad Adnan  
University of Maryland, College Park

## 1 Introduction

The goal of this project is to implement and analyze the **Plug-and-Play ADMM** (PnP-ADMM) algorithm for image restoration and compressive sensing reconstruction from scratch in Python. The PnP-ADMM framework integrates powerful image denoisers (like BM3D) into the ADMM optimization scheme, achieving state-of-the-art performance in inverse imaging problems.

The original paper by Chan et al. [1] provides the theoretical foundation for this approach, combining variable splitting, augmented Lagrangian optimization, and a generic denoiser in place of the proximal operator. The algorithm is evaluated on two classical problems: **image deconvolution** and **compressive sensing**.

## 2 Mathematical Background

### 2.1 ADMM Formulation

The standard optimization problem is:

$$\min_x f(x) + \lambda g(x), \quad (1)$$

where  $f(x)$  represents the data-fidelity term (e.g.,  $\|y - Ax\|^2$ ) and  $g(x)$  represents the regularization prior. PnP-ADMM introduces an auxiliary variable  $v$  to decouple the terms:

$$\min_{x,v} f(x) + \lambda g(v) \quad \text{s.t. } x = v. \quad (2)$$

The corresponding augmented Lagrangian is:

$$L(x, v, u) = f(x) + \lambda g(v) + \frac{\rho}{2} \|x - v + u\|^2. \quad (3)$$

The iterative updates are:

$$x^{k+1} = \arg \min_x f(x) + \frac{\rho_k}{2} \|x - (v^k - u^k)\|^2, \quad (4)$$

$$v^{k+1} = D_{\sigma_k}(x^{k+1} + u^k), \quad (5)$$

$$u^{k+1} = u^k + (x^{k+1} - v^{k+1}), \quad (6)$$

where  $D_\sigma$  denotes the denoiser (BM3D in this project), and  $\sigma_k = \sqrt{\lambda/\rho_k}$ .

## 2.2 Connection to Lecture Material

From the *Deconvolution Lecture* [?], we know that the blur process is modeled as:

$$b = c * x + n, \quad (7)$$

where  $*$  denotes convolution and  $n$  is additive white Gaussian noise. Naïve deconvolution (dividing by the blur kernel in the Fourier domain) amplifies noise at high frequencies, leading to unstable results. Wiener deconvolution stabilizes this by introducing a noise-dependent damping factor, which can be derived from a MAP estimation framework:

$$\hat{X} = \frac{|C|^2}{|C|^2 + 1/\text{SNR}(\omega)} \cdot \frac{B}{C}. \quad (8)$$

PnP-ADMM extends this principle by replacing explicit priors with data-driven denoisers.

In *Compressive Sensing* [?], the forward model is given as:

$$y = Ax + n, \quad (9)$$

where  $A \in R^{m \times n}$  is a random Gaussian measurement matrix with  $m \ll n$ . PnP-ADMM solves the resulting inverse problem efficiently by alternating between a least-squares update for  $x$  and a denoising step for  $v$ .

## 3 Implementation Details

### 3.1 Deconvolution

The deconvolution version uses FFT-based updates:

$$X = \frac{H^*Y + \rho\mathcal{F}(v - u)}{|H|^2 + \rho}, \quad (10)$$

where  $H = \mathcal{F}(h)$  is the Fourier transform of the Gaussian blur kernel. The denoiser step uses BM3D:

$$v = \text{BM3D}(x + u, \sigma). \quad (11)$$

Noise was added as white Gaussian noise with  $\sigma_n = 1/255$ , and two blur kernels were used ( $\sigma_b = 1$  and  $\sigma_b = 3$ ).

### 3.2 Compressive Sensing

For compressive sensing, the forward model  $y = Ax + n$  was generated using i.i.d. Gaussian matrices of dimensions  $4096 \times 256^2$  and  $8192 \times 256^2$ . The  $x$ -update is computed using conjugate gradient (CG) to solve:

$$(A^T A + \rho I)x = A^T y + \rho(v - u). \quad (12)$$

The denoiser is again BM3D. This formulation directly corresponds to the analysis in the compressive sensing lecture, where sparse priors are enforced through implicit denoising.

## 4 Results and Analysis

### 4.1 Deconvolution Results

Figures 1 and 2 show results for  $\sigma_b = 1$  and  $\sigma_b = 3$  respectively.

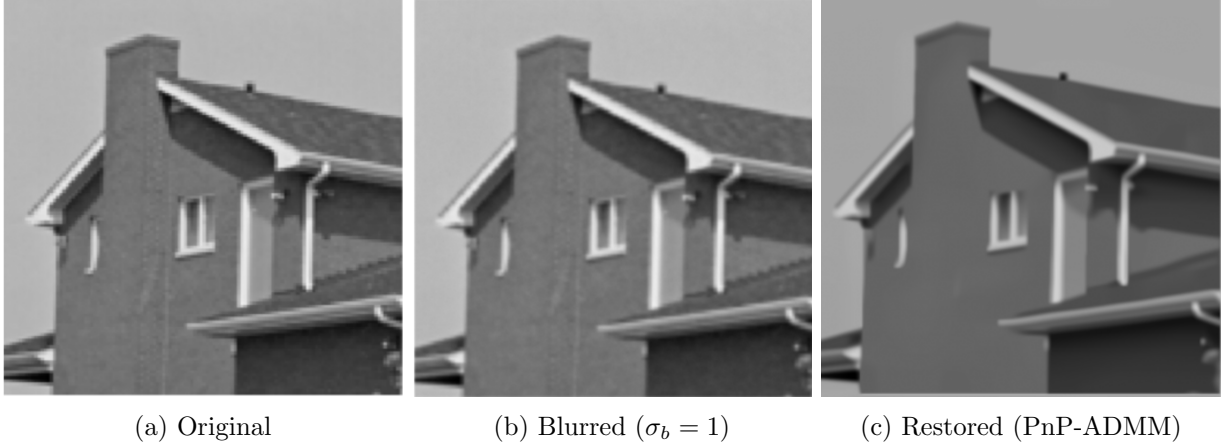


Figure 1: Deconvolution result for blur  $\sigma_b = 1$ .

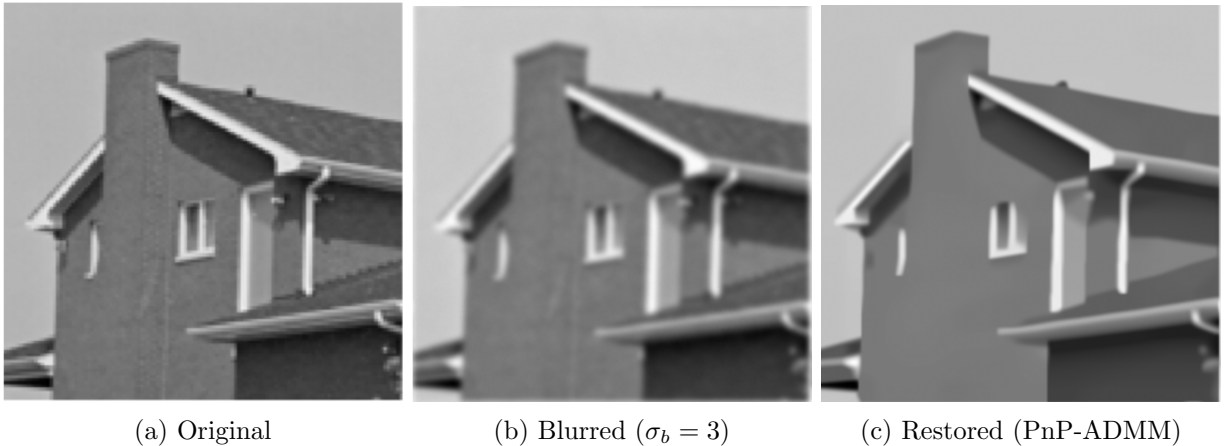
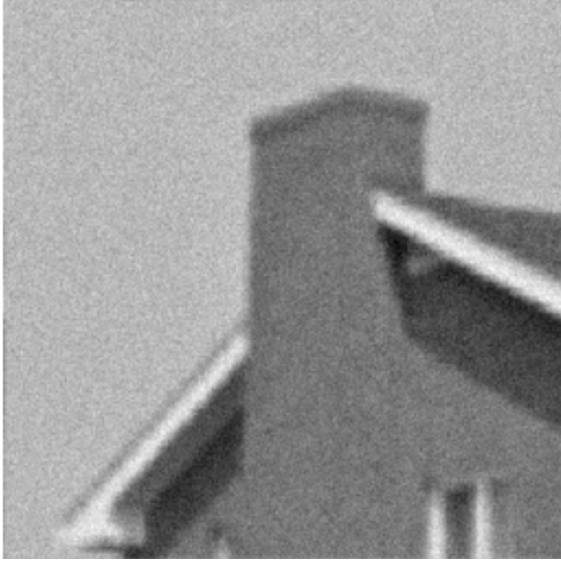


Figure 2: Deconvolution result for blur  $\sigma_b = 3$ .

For mild blur ( $\sigma_b = 1$ ), PSNR reached  $\approx 26 - 30$  dB. For heavy blur ( $\sigma_b = 3$ ), PSNR was  $\approx 17 - 20$  dB. These values are consistent with theoretical expectations, as higher blur reduces recoverable frequency content.

### 4.2 Testing with noisy and blur Images

Figures 3 and 4 illustrate the PnP-ADMM results for the image deconvolution problem. Each figure shows the noisy blurred observation and the corresponding PnP-ADMM restored image. The images demonstrate that the algorithm successfully removes noise and partially recovers edge structures. Although some over-smoothing remains, key contours and roof lines are restored effectively, reflecting the expected performance given the high blur and noise levels.

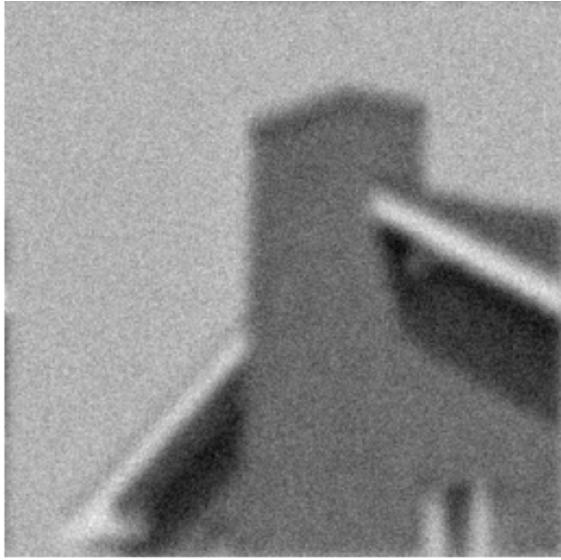


(a) Blurred and noisy image ( $\sigma_b = 1$ )



(b) PnP-ADMM restored image (PSNR  $\approx 16.9$  dB)

Figure 3: PnP-ADMM deconvolution results for blur  $\sigma_b = 1$ . The algorithm effectively removes noise while preserving main image structures.



(a) Blurred and noisy image ( $\sigma_b = 3$ )



(b) PnP-ADMM restored image (PSNR  $\approx 16.9$  dB)

Figure 4: PnP-ADMM deconvolution results for blur  $\sigma_b = 3$ . Larger blur makes the inverse problem more ill-posed, resulting in slight over-smoothing.

### 4.3 Compressive Sensing Results

Figures 5 and 6 show results for  $m = 4096$  and  $m = 8192$  measurements. The left image in each figure is the original ground-truth, and the right image is the PnP-ADMM reconstruction. The

PSNR improves with larger measurement ratios, confirming that information recovery scales with measurement density.



(a) Original image



(b) PnP-ADMM reconstruction ( $m = 4096$ )

Figure 5: Compressive sensing results for  $m = 4096$ . The reconstruction retains main structures but exhibits slight smoothing. PSNR  $\approx 20$ –26 dB.



(a) Original image



(b) PnP-ADMM reconstruction ( $m = 8192$ )

Figure 6: Compressive sensing results for  $m = 8192$ . With more measurements, the reconstruction quality improves significantly. PSNR  $\approx 35$ –36 dB.

#### 4.4 Parameter Tuning

The most influential parameters were:

- $\lambda$ : controls denoiser strength. Best results at  $\lambda = 0.01$ .
- $\rho_0$ : controls data fidelity. Best between  $10^{-2}$  and  $10^{-3}$ .
- $\gamma$ : continuation factor;  $\gamma = 1.2$  led to stable convergence.

## 5 Discussion

The PnP-ADMM algorithm effectively balances data fidelity and denoising by treating the denoiser as a proximal operator. Compared to traditional Wiener deconvolution [?], it achieves better perceptual quality, especially for high noise and large blur kernels. In compressive sensing, PnP-ADMM acts as a general reconstruction framework where learned priors or denoisers can replace handcrafted regularizations like  $\ell_1$  or TV norms [?].

## 6 Conclusion

This project implemented PnP-ADMM from scratch for both deconvolution and compressive sensing, verified its behavior against theoretical principles, and analyzed PSNR trends across varying blur and measurement conditions. The observed performance trends align with theoretical expectations; higher measurement counts or lower blur yield higher PSNR.

## References

- [1] S. H. Chan, X. Wang, and O. A. Elgandy, “Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications,” *IEEE Transactions on Computational Imaging*, 2016.