Plug-and-Play Methods Provably Converge with Properly Trained Denoisers

Ernest K. Ryu[1] Jialin Liu[1] Sicheng Wang[2] Xiaohan Chen[2]
Zhangyang Wang[2] Wotao Yin[1]

2019 International Conference on Machine Learning

[1] UCLA Mathematics
[2] Texas A&M Computer Science and Engineering

Image processing via optimization

Consider recovering or denoising an image through the optimization

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x) + \gamma g(x),$$

- x is image
- $f(x)$ is data fidelity (a posteriori knowledge)
- $g(x)$ is noisiness of the image (a priori knowledge)
- $\gamma \geq 0$ is relative importance between f and g

Image processing via ADMM

We often use first-order methods, such as ADMM

$$
\begin{aligned}
x^{k+1} &= \operatorname*{argmin}_{x \in \mathbb{R}^d} \left\{ \sigma^2 g(x) + (1/2)\|x - (y^k - u^k)\|^2 \right\} \\
y^{k+1} &= \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ \alpha f(y) + (1/2)\|y - (x^{k+1} + u^k)\|^2 \right\} \\
u^{k+1} &= u^k + x^{k+1} - y^{k+1}
\end{aligned}
$$

with $\sigma^2 = \alpha\gamma$.

Image processing via ADMM

More concise notation

$$\begin{aligned} x^{k+1} &= \mathrm{Prox}_{\sigma^2 g}(y^k - u^k) \\ y^{k+1} &= \mathrm{Prox}_{\alpha f}(x^{k+1} + u^k) \\ u^{k+1} &= u^k + x^{k+1} - y^{k+1}. \end{aligned}$$

The proximal operator of h is

$$\mathrm{Prox}_{\alpha h}(z) = \operatorname*{argmin}_{x \in \mathbb{R}^d} \left\{ \alpha h(x) + (1/2)\|x - z\|^2 \right\}.$$

(Well-defined if h is proper, closed, and convex.)

Interpretations of ADMM subroutines

The subroutine $\mathrm{Prox}_{\sigma^2 g} : \mathbb{R}^d \to \mathbb{R}^d$ is a denoiser, i.e.,

$$\mathrm{Prox}_{\sigma^2 g} : \text{noisy image} \mapsto \text{less noisy image}$$

$\mathrm{Prox}_{\alpha f} : \mathbb{R}^d \to \mathbb{R}^d$ enforces consistency with measured data, i.e.,

$$\mathrm{Prox}_{\alpha f} : \text{less consistent} \mapsto \text{more consistent with data}$$

Other denoisers

However, some state-of-the-art image denoisers do not originate from optimization problems. (E.g. NLM, BM3D, and CNN.) Nevertheless, such a denoiser $H_\sigma : \mathbb{R}^d \to \mathbb{R}^d$ still has the interpretation

$$H_\sigma : \text{noisy image} \mapsto \text{less noisy image}$$

where $\sigma \geq 0$ is a noise parameter.

It is possible to integrate such denoisers with existing algorithms such as ADMM or proximal gradient?

Plug and play!

To address this question, Venkatakrishnan et al.[3] proposed Plug-and-Play ADMM (PnP-ADMM), which simply replaces the proximal operator $\mathrm{Prox}_{\sigma^2 g}$ with the denoiser H_σ:

$$\begin{aligned} x^{k+1} &= H_\sigma(y^k - u^k) \\ y^{k+1} &= \mathrm{Prox}_{\alpha f}(x^{k+1} + u^k) \\ u^{k+1} &= u^k + x^{k+1} - y^{k+1}. \end{aligned}$$

Surprisingly and remarkably, this ad-hoc method exhibited great empirical success, and spurred much follow-up work.

[3] Venkatakrishnan, Bouman, and Wohlberg, Plug-and-play priors for model based reconstruction, IEEE GlobalSIP, 2013.

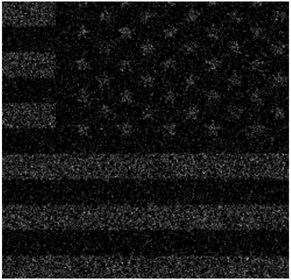
Plug and play!

By integrating modern denoising priors into ADMM or other proximal algorithms, PnP combines the advantages of data-driven operators and classic optimization.

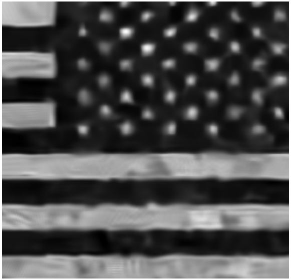
In image denoising, PnP replaces total variation regularization with an explicit denoiser such as BM3D or deep learning-based denoisers.

PnP is suitable when end-to-end training is impossible (e.g. due to insufficient data or time).

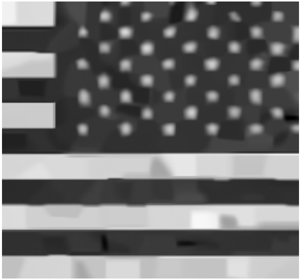
Example: Poisson denoising



Corrupted image



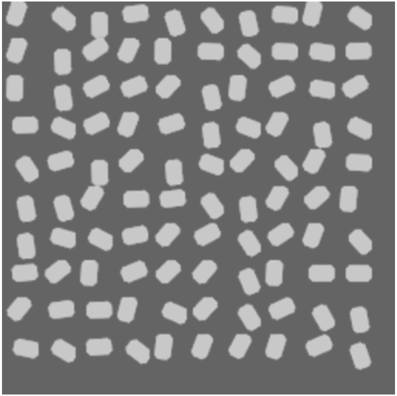
Other method



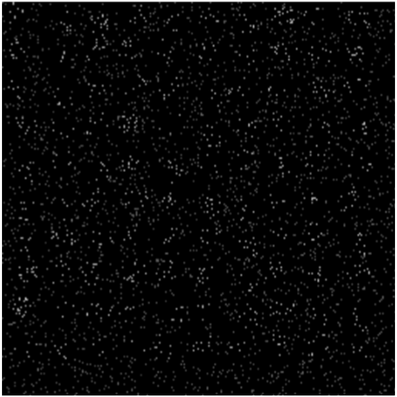
PnP-ADMM with BM3D

Rond, Giryes, and Elad, J. Vis. Commun. Image R. 2016.

Example: Inpainting



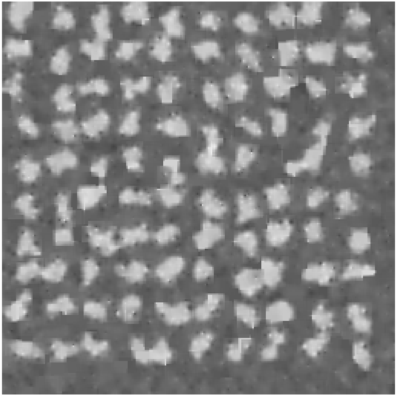
Original image



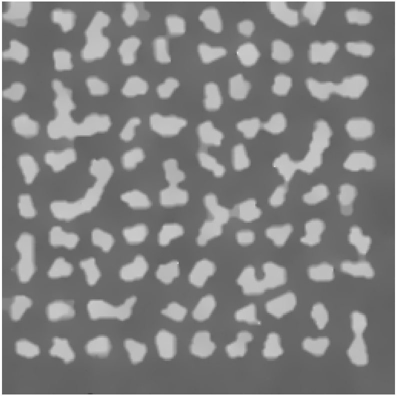
5% random sampling

Sreehari et al., IEEE Trans. Comput. Imag., 2016.

Example: Inpainting



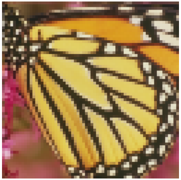
Other method



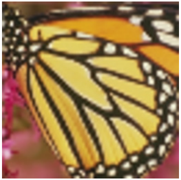
PnP-ADMM with NLM

Sreehari et al., IEEE Trans. Comput. Imag., 2016.

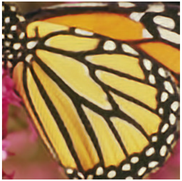
Example: Super resolution



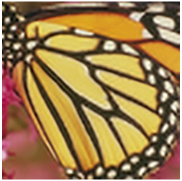
Low resolution input



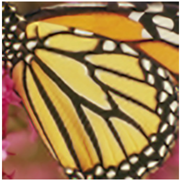
Other method



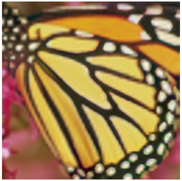
Other method



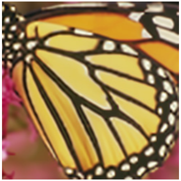
Other method



Other method



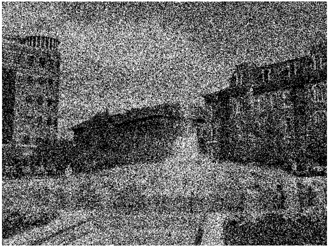
Other method



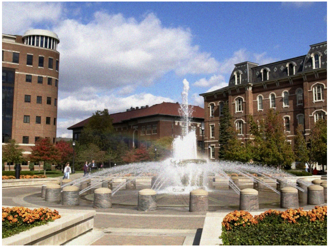
PnP-ADMM with BM3D

Chan, Wang, Elgendy, IEEE Trans. Comput. Imag., 2017.

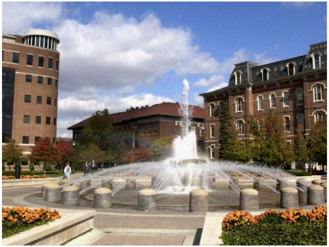
Example: Single photon imaging



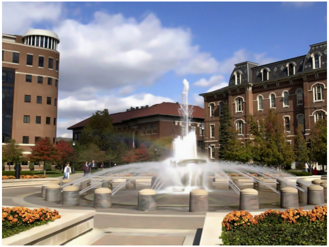
Corrupted image



other method



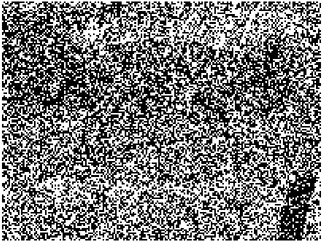
other method



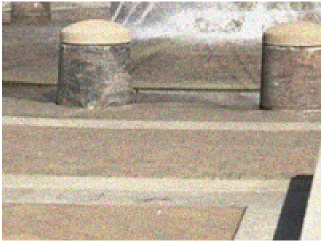
PnP-ADMM with BM3D

Chan, Wang, Elgendy, IEEE Trans. Comput. Imag., 2017.

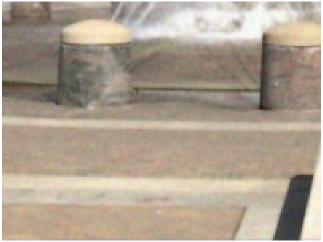
Example: Single photon imaging



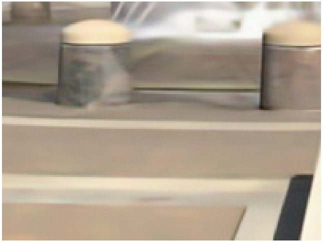
Corrupted image



other method



other method



PnP-ADMM with BM3D

Chan, Wang, Elgendy, IEEE Trans. Comput. Imag., 2017.

Contribution of this work

The empirical success of Plug-and-Play (PnP) naturally leads us to ask theoretical questions: **When does PnP converge and what denoisers can we use?**

- We prove convergence of PnP methods under a certain Lipschitz condition.
- We propose real spectral normalization, a technique for constraining deep learning-based denoisers in their training to enforce the proposed Lipschitz condition.
- We present experimental results validating our theory.[4]

[4] Code available at: `https://github.com/uclaopt/Provable_Plug_and_Play/`

Outline

PNP-FBS/ADMM and their fixed points

Convergence via contraction

Real spectral normalization: Enforcing Assumption (A)

Experimental validation

PnP FBS

Plug-and-play forward-backward splitting:

$$x^{k+1} = H_\sigma(I - \alpha \nabla f)(x^k) \tag{PNP-FBS}$$

where $\alpha > 0$.

PnP FBS

PNP-FBS is a fixed-point iteration, and $x^\star$ is a fixed point if

$$x^\star = H_\sigma(I - \alpha\nabla f)(x^\star).$$

Interpretation of fixed points: A compromise between making the image agree with measurements and making the image less noisy.

PnP ADMM

Plug-and-play alternating directions method of multipliers:

$$\begin{aligned} x^{k+1} &= H_\sigma(y^k - u^k) \\ y^{k+1} &= \mathrm{Prox}_{\alpha f}(x^{k+1} + u^k) \\ u^{k+1} &= u^k + x^{k+1} - y^{k+1} \end{aligned} \qquad \text{(PNP-ADMM)}$$

where $\alpha > 0$.

PnP ADMM

PNP-ADMM is a fixed-point iteration, and $(x^\star, u^\star)$ is a fixed point if

$$\begin{aligned} x^\star &= H_\sigma(x^\star - u^\star) \\ x^\star &= \mathrm{Prox}_{\alpha f}(x^\star + u^\star). \end{aligned}$$

PnP DRS

Plug-and-play Douglas–Rachford splitting:

$$\begin{aligned} x^{k+1/2} &= \mathrm{Prox}_{\alpha f}(z^k) \\ x^{k+1} &= H_\sigma(2x^{k+1/2} - z^k) \\ z^{k+1} &= z^k + x^{k+1} - x^{k+1/2} \end{aligned} \qquad \text{(PNP-DRS)}$$

where $\alpha > 0$.

We can write PNP-DRS as $z^{k+1} = T(z^k)$ with

$$T = \frac{1}{2}I + \frac{1}{2}(2H_\sigma - I)(2\mathrm{Prox}_{\alpha f} - I).$$

PNP-ADMM and PNP-DRS are equivalent. We analyze convergence of PNP-DRS and translate the result to PNP-ADMM.

PnP DRS

PNP-DRS is a fixed-point iteration, and $z^\star$ is a fixed point if

$$\begin{aligned} x^\star &= \mathrm{Prox}_{\alpha f}(z^\star) \\ x^\star &= H_\sigma(2x^\star - z^\star). \end{aligned}$$

Outline

PNP-FBS/ADMM and their fixed points

Convergence via contraction

Real spectral normalization: Enforcing Assumption (A)

Experimental validation

What we do not assume

If we assume $2H_\sigma - I$ is nonexpansive, standard tools of monotone operator theory tell us that PnP-ADMM converges. However, this assumption is unrealistic[5] so we do not assume it.

We do not assume H_σ is continuously differentiable.

[5] Chan, Wang, and Elgendy, Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications, IEEE TCI, 2017.

Main assumption

Rather, we assume $H_\sigma : \mathbb{R}^d \to \mathbb{R}^d$ satisfies

$$\|(H_\sigma - I)(x) - (H_\sigma - I)(y)\| \leq \varepsilon \|x - y\| \tag{A}$$

for all $x, y \in \mathbb{R}^d$ for some $\varepsilon \geq 0$. Since σ controls the strength of the denoising, we can expect H_σ to be close to identity for small σ. If so , Assumption (A) is reasonable.

Contractive operators

Under (A), we show PNP-FBS and PNP-DRS are **contractive** iterations in the sense that we can express the iterations as $x^{k+1} = T(x^k)$, where $T : \mathbb{R}^d \to \mathbb{R}^d$ satisfies

$$\|T(x) - T(y)\| \leq \delta\|x - y\|$$

for all $x, y \in \mathbb{R}^d$ for some $\delta < 1$.

If $x^\star$ satisfies $T(x^\star) = x^\star$, i.e., $x^\star$ is a fixed point, then $x^k \to x^\star$ geometrically by the classical Banach contraction principle.

Convergence of PNP-FBS

Theorem

Assume H_σ *satisfies assumption* (A) *for some* $\varepsilon \geq 0$. *Assume* f *is* μ*-strongly convex,* f *is differentiable, and* ∇f *is* L*-Lipschitz. Then*

$$T = H_\sigma(I - \alpha \nabla f)$$

satisfies

$$\|T(x) - T(y)\| \leq \max\{|1 - \alpha\mu|, |1 - \alpha L|\}(1 + \varepsilon)\|x - y\|$$

for all $x, y \in \mathbb{R}^d$. *The coefficient is less than* 1 *if*

$$\frac{1}{\mu(1 + 1/\varepsilon)} < \alpha < \frac{2}{L} - \frac{1}{L(1 + 1/\varepsilon)}.$$

Such an α *exists if* $\varepsilon < 2\mu/(L - \mu)$.

Convergence of PNP-DRS

Theorem

Assume H_σ satisfies assumption (A) *for some $\varepsilon \geq 0$. Assume f is μ-strongly convex and differentiable. Then*

$$T = \frac{1}{2}I + \frac{1}{2}(2H_\sigma - I)(2\mathrm{Prox}_{\alpha f} - I)$$

satisfies

$$\|T(x) - T(y)\| \leq \frac{1 + \varepsilon + \varepsilon\alpha\mu + 2\varepsilon^2\alpha\mu}{1 + \alpha\mu + 2\varepsilon\alpha\mu}\|x - y\|$$

for all $x, y \in \mathbb{R}^d$. The coefficient is less than 1 if

$$\frac{\varepsilon}{(1 + \varepsilon - 2\varepsilon^2)\mu} < \alpha, \quad \varepsilon < 1.$$

Convergence of PNP-ADMM

Corollary

Assume H_σ satisfies assumption (A) *for some $\varepsilon \in [0,1)$. Assume f is μ-strongly convex. Then PNP-ADMM converges for*

$$\frac{\varepsilon}{(1+\varepsilon-2\varepsilon^2)\mu} < \alpha.$$

PnP-FBS vs. PnP-ADMM

PNP-FBS and PNP-ADMM share the same fixed points [6] [7]. They are distinct methods for finding the same set of fixed points.

PNP-FBS is easier to implement as it requires ∇f rather than $\mathrm{Prox}_{\alpha f}$.

PNP-ADMM has better convergence properties as demonstrated by Theorems 1 and 2 and our experiments.

[6] Meinhardt, Moeller, Hazirbas, and Cremers, Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. ICCV, 2017.

[7] Sun, Wohlberg, and Kamilov, An online plug-and-play algorithm for regularized image reconstruction. IEEE TCI, 2019.

Convergence proof sketch

PnP-FBS: The iteration is composition of an expansive operator with a contractive operator.

PnP-DRS: Proof is based on the notion "negatively averaged" operators of Giselsson [8].

[8] Giselsson, Tight global linear convergence rate bounds for Douglas–Rachford splitting, J. Fix. Point. Theory. Appl., 2017

Outline

PNP-FBS/ADMM and their fixed points

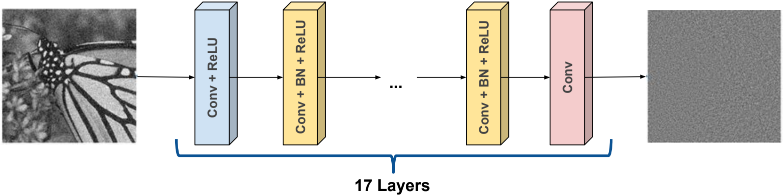
Convergence via contraction

Real spectral normalization: Enforcing Assumption (A)

Experimental validation

Deep learning denoiser: DnCNN

We use DnCNN[9], which learns the residual mapping with a 17-layer CNN.

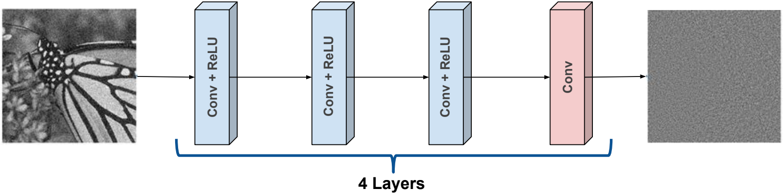


Given a noisy observation $y = x + e$, where x is the clean image and e is noise, the residual mapping R outputs the noise, i.e., $R(y) = e$ so that $y - R(y)$ is the clean recovery. Learning the residual mapping is a common approach in deep learning-based image restoration.

[9] Zhang, Zuo, Chen, Meng, and Zhang, Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising, IEEE TIP, 2017.

Deep learning denoiser: SimpleCNN

We also construct a simple convolutional encoder-decoder model for denoising and call it SimpleCNN.



We use SimpleCNN to show realSN is applicable to any CNN denoiser.

Lipschitz constrained deep denoising

Note

$$(I - H_\sigma)(y) = y - H_\sigma(y) = R(y),$$

with denoiser H_σ, residual R, and identity I.

Enforcing

$$\|(I - H_\sigma)(x) - (I - H_\sigma)(y)\| \leq \varepsilon \|x - y\| \tag{A}$$

is equivalent to constraining the Lipschitz constant of R. We propose a variant of the spectral normalization for this.

Spectral normalization

Miyato et al.[10] proposed spectral normalization (SN), which controls the Lipschitz constant of a network's layers through controlling the spectral norm of the layer's weight. If we use 1-Lipschitz nonlinearities (such as ReLU), the Lipschitz constant of a layer is upper-bounded by the spectral norm of its weight, and the Lipschitz constant of the full network is bounded by the product of spectral norms of all layers.

While this basic methodology suits our goal, Miyato et al.'s SN uses an inexact implementation that underestimates the true spectral norm.

[10] Miyato, Kataoka, Koyama, and Yoshida, Spectral Normalization for Generative Adversarial Networks, ICLR, 2018.

Real Spectral Normalization

Real Spectral Normalization (realSN) accurately constrains the network's Lipschitz constant through a power iteration with the convolutional linear operator $\mathcal{K}_l : \mathbb{R}^{C_{\text{in}} \times h \times w} \to \mathbb{R}^{C_{\text{out}} \times h \times w}$, where h, w are input's height and width, and its conjugate (transpose) operator $\mathcal{K}_l^*$. The iteration maintains $U_l \in \mathbb{R}^{C_{\text{out}} \times h \times w}$ and $V_l \in \mathbb{R}^{C_{\text{in}} \times h \times w}$ to estimate the leading left and right singular vectors respectively. During each forward pass of the neural network, realSN conducts:

1. Apply one step of the power method with operator $\mathcal{K}_l$:

$$\begin{aligned} V_l &\leftarrow \mathcal{K}_l^*(U_l) \ / \ \|\mathcal{K}_l^*(U_l)\|_2, \\ U_l &\leftarrow \mathcal{K}_l(V_l) \ / \ \|\mathcal{K}_l(V_l)\|_2. \end{aligned}$$

2. Normalize the convolutional kernel K_l with estimated spectral norm:

$$K_l \leftarrow K_l / \sigma(\mathcal{K}_l), \text{ where } \sigma(\mathcal{K}_l) = \langle U_l, \mathcal{K}_l(V_l) \rangle$$

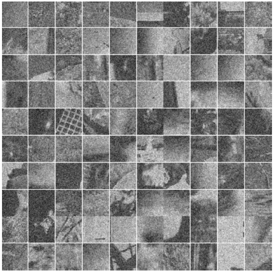
We can view realSN as an approximate projected gradient enforcing the Lipschitz continuity constraint.

Implementation details

We train SimpleCNN and DnCNN in the setting of Gaussian denoising with 40×40 patches of the BSD500 dataset, natural images. RealSN constrains the Lipschitz constant to no more than 1.

BSD500
original images

40×40
(clean) patches



40×40 patches
corrupted with
Gaussian noise

On an Nvidia GTX 1080 Ti, DnCNN took 4.08 hours and realSN-DnCNN took 5.17 hours to train, so the added cost of realSN is mild.

Outline

PNP-FBS/ADMM and their fixed points

Convergence via contraction

Real spectral normalization: Enforcing Assumption (A)

Experimental validation

Poisson denoising

Given a true image $x_{\mathrm{true}} \in \mathbb{R}^d$, we observe Poisson random variables

$$y_i \sim \mathsf{Poisson}((x_{\mathrm{true}})_i)$$

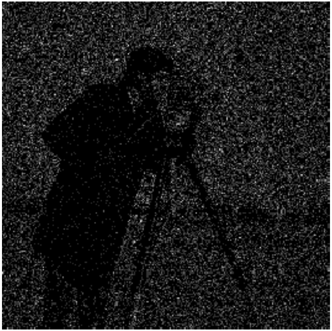
for $i = 1, \ldots, d$. We use the negative log-likelihood

$$f(x) = \sum_{i=1}^{d} -y_i \log(x_i) + x_i.$$

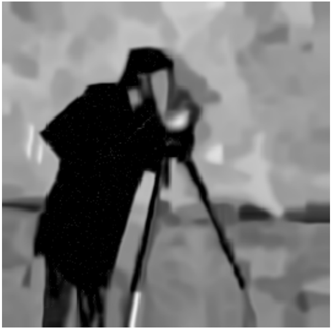
For further details of the experimental setup, see the main paper or [11].

[11] Rond, Giryes, and Elad, Poisson inverse problems by the plug-and-play scheme, J. Vis. Commun. Image R. 2016.

Poisson denoising

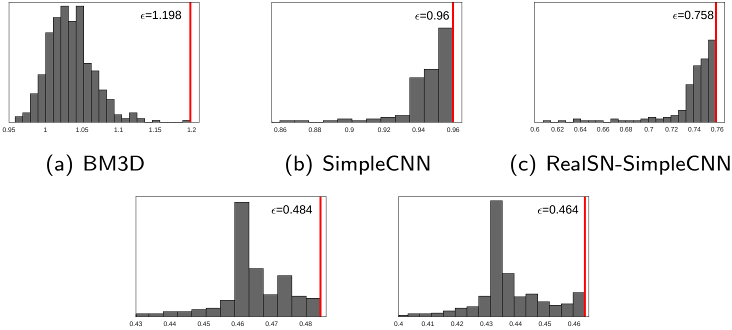


Corrupted 3.36dB



Recovery 20.28dB

Poisson denoising



(a) BM3D

(b) SimpleCNN

(c) RealSN-SimpleCNN

(d) DnCNN

(e) RealSN-DnCNN

We run PnP iterations, calculate $\|(I - H_\sigma)(x) - (I - H_\sigma)(y)\|/\|x - y\|$ between the iterates and the limit, and plot the histogram. The maximum value, the red bar, lower-bounds ε of (A). Convergence of PnP-ADMM requires $\varepsilon < 1$. The results prove BM3D violates this assumption and illustrate that RealSN indeed controls (reduces) the Lipschitz constant.

Poisson denoising

	BM3D	RealSN-DnCNN	RealSN-SimpleCNN
PNP-ADMM	23.4617	**23.5873**	18.7890
PNP-FBS	18.5835	22.2154	**22.7280**

PSNR of the PnP methods with BM3D, RealSN-DnCNN, and RealSN-SimpleCNN plugged in. In both PnP methods, one of the two denoisers using RealSN, for which we have theory, outperforms BM3D.

Single photon imaging

The measurement model of quanta image sensors is

$$z = \mathbf{1}(y \geq 1), \quad y \sim \mathsf{Poisson}(\alpha_{sg} G x_{\text{true}})$$

where $x_{\text{true}} \in \mathbb{R}^d$ is the true image, $G : \mathbb{R}^d \to \mathbb{R}^{dK}$ duplicates each pixel to K pixels, $\alpha_{sg} \in \mathbb{R}$ is sensor gain, K is the oversampling rate, $z \in \{0, 1\}^{dK}$ is the observed binary photons. (y is not measured.) The likelihood function is

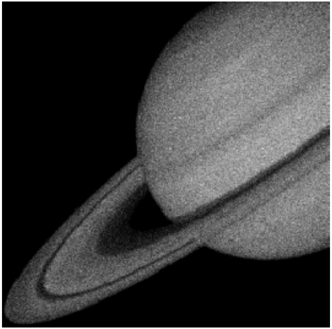
$$f(x) = \sum_{j=1}^{n} -K_j^0 \log(e^{-\alpha_{sg} x_j / K}) - K_j^1 \log(1 - e^{-\alpha_{sg} x_j / K}),$$

where K_j^1 is the number of ones in the j-th unit pixel, K_j^0 is the number of zeros in the j-th unit pixel.

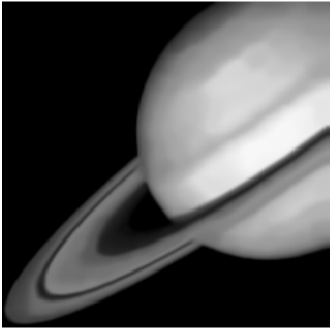
For further details of the experimental setup, see the main paper or [12].

[12] Elgendy and Chan, Image reconstruction and threshold design for quanta image sensors, IEEE ICIP, 2016.

Single photon imaging



Corrupted 17.32dB



Recovery 36.02dB

Measurement pixels take integer values between 0 and $K = 64$.

Single photon imaging

PnP-ADMM with RealSN-DnCNN provides best PSNR. We also observe that RealSN makes PnP converge more stably.

PnP-FBS, $\alpha = 0.005$

Average PSNR	BM3D	RealSN-DnCNN	RealSN-SimpleCNN
Iteration 50	28.7933	27.9617	29.0062
Iteration 100	29.0510	27.9887	29.0517
Best Overall	**29.5327**	28.4065	29.3563

PnP-ADMM, $\alpha = 0.01$

Average PSNR	BM3D	RealSN-DnCNN	RealSN-SimpleCNN
Iteration 50	30.0034	31.0032	29.2154
Iteration 100	30.0014	31.0032	29.2151
Best Overall	30.0474	**31.0431**	29.2155

Compressed sensing MRI

PnP is useful in medical imaging when we do not have enough data for end-to-end training: train the denoiser H_σ on natural images, and “plug” it into the PnP framework to be applied to medical images.

Given a true image $x_{\mathrm{true}} \in \mathbb{C}^d$, CS-MRI measures

$$y = \mathcal{F}_p x_{\mathrm{true}} + \varepsilon_e,$$

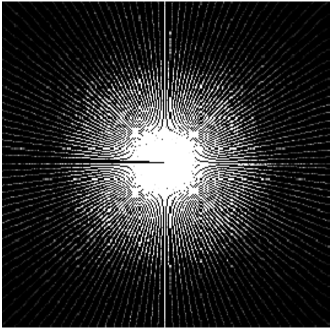
where $\mathcal{F}_p$ is the Fourier k-domain subsampling (partial Fourier operator), and $\varepsilon_e \sim N(0, \sigma_e I_k)$ is measurement noise. We use the objective function

$$f(x) = (1/2)\|y - \mathcal{F}_p x\|^2.$$

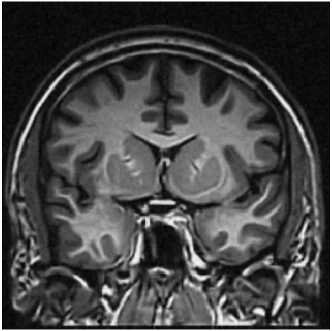
For further details of the experimental setup, see the main paper or [13].

[13] Eksioglu, Decoupled algorithm for MRI reconstruction using nonlocal block matching model: BM3D-MRI, J. Math. Imaging Vis., 2016.

Compressed sensing MRI



Radial sampling k-space



Recovery 19.09dB

k-space measurement is complex-valued so we plot the absolute value.

Compressed sensing MRI

PSNR (in dB) for 30% sampling with additive Gaussian noise $\sigma_e = 15$. RealSN generally improves the performance.

Sampling approach		Random		Radial		Cartesian	
Image		Brain	Bust	Brain	Bust	Brain	Bust
Zero-filling		9.58	7.00	9.29	6.19	8.65	6.01
TV[14]		16.92	15.31	15.61	14.22	12.77	11.72
RecRF[15]		16.98	15.37	16.04	14.65	12.78	11.75
BM3D-MRI[16]		17.31	13.90	16.95	13.72	14.43	12.35
PnP-FBS	BM3D	19.09	16.36	18.10	15.67	14.37	12.99
	DnCNN	19.59	16.49	18.92	15.99	14.76	14.09
	RealSN-DnCNN	**19.82**	**16.60**	**18.96**	**16.09**	**14.82**	**14.25**
	SimpleCNN	15.58	12.19	15.06	12.02	12.78	10.80
	RealSN-SimpleCNN	17.65	14.98	16.52	14.26	13.02	11.49
PnP-ADMM	BM3D	19.61	**17.23**	18.94	**16.70**	14.91	13.98
	DnCNN	19.86	17.05	19.00	16.64	14.86	14.14
	RealSN-DnCNN	**19.91**	17.09	**19.08**	16.68	**15.11**	**14.16**
	SimpleCNN	16.68	12.56	16.83	13.47	13.03	11.17
	RealSN-SimpleCNN	17.77	14.89	17.00	14.47	12.73	11.88

[14] Lustig, Santos, Lee, Donoho, and Pauly, SPARS, 2005.

[15] Yang, Zhang, and Yin, IEEE JSTSP, 2010.

[16] Eksioglu, J. Math. Imaging Vis., 2016.

Conclusion

1. PnP-FBS and PnP-ADMM converges under a Lipschitz assumption on the denoiser.
2. Real spectral normalization enforces the Lipschitz condition in training deep learning-based denoisers.
3. The experiments validate the theory.

Paper available at:
`http://proceedings.mlr.press/v97/ryu19a.html`
Code available at:
`https://github.com/uclaopt/Provable_Plug_and_Play/`

Link to paper

Link to code