

Proximal Denoiser for Convergent Plug-and-Play Optimization with Nonconvex Regularization

Samuel Hurault, Arthur Leclaire, Nicolas Papadakis

Institut de Mathématiques de Bordeaux



Image Inverse Problems

Find x from observation $y = Ax + \xi$

- $y \in \mathbb{R}^m$ observation
- $x \in \mathbb{R}^n$ unknown input
- $A \in \mathbb{R}^{m \times n}$ degradation operator
- ξ random noise, generally $\xi \sim \mathcal{N}(0, \sigma^2 \text{Id}_m)$

Image Inverse Problems

Find x from observation $y = Ax + \xi$

- $y \in \mathbb{R}^m$ observation
- $x \in \mathbb{R}^n$ unknown input
- $A \in \mathbb{R}^{m \times n}$ degradation operator
- ξ random noise, generally $\xi \sim \mathcal{N}(0, \sigma^2 \text{Id}_m)$

Denoising:



Image Inverse Problems

Find x from observation $y = Ax + \xi$

- $y \in \mathbb{R}^m$ observation
- $x \in \mathbb{R}^n$ unknown input
- $A \in \mathbb{R}^{m \times n}$ degradation operator
- ξ random noise, generally $\xi \sim \mathcal{N}(0, \sigma^2 \text{Id}_m)$

Deblurring:

$$\text{Butterfly Image} = \text{Blur Kernel} * \text{Clean Image} + \text{Noise}$$


Image Inverse Problems

Find x from observation $y = Ax + \xi$

- $y \in \mathbb{R}^m$ observation
- $x \in \mathbb{R}^n$ unknown input
- $A \in \mathbb{R}^{m \times n}$ degradation operator
- ξ random noise, generally $\xi \sim \mathcal{N}(0, \sigma^2 \text{Id}_m)$

Super-resolution:

$$\text{Butterfly Image} = \left(\begin{matrix} \text{Small Image} \\ * \\ \text{Large Image} \end{matrix} \right) \Downarrow_s + \text{Noise}$$

Image Inverse Problems

Find x from observation $y = Ax + \xi$

- $y \in \mathbb{R}^m$ observation
- $x \in \mathbb{R}^n$ unknown input
- $A \in \mathbb{R}^{m \times n}$ degradation operator
- ξ random noise, generally $\xi \sim \mathcal{N}(0, \sigma^2 \text{Id}_m)$

Inpainting:

$$\begin{matrix} \text{Butterfly wing image} & = & \text{Noise matrix} & \otimes & \text{Butterfly wing image} \end{matrix}$$

Maximum A-Posteriori

Find x from observation $y = Ax + \xi$

Maximum A-Posteriori

$$x^* \in \arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

Maximum A-Posteriori

Find x from observation $y = Ax + \xi$

Maximum A-Posteriori

$$x^* \in \arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

$$\iff \arg \min_{x \in \mathbb{R}^n} \text{data-fidelity}$$

$$\text{e.g. } f(x) = \frac{1}{2} \|Ax - y\|^2$$

Maximum A-Posteriori

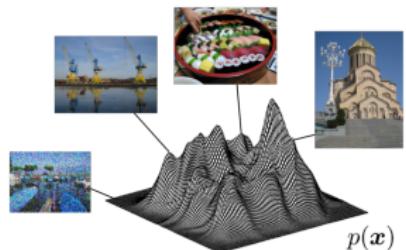
Find x from observation $y = Ax + \xi$

Maximum A-Posteriori

$$x^* \in \arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

$$\iff \arg \min_{x \in \mathbb{R}^n} \text{data-fidelity} + \text{prior}$$

$$\text{e.g. } f(x) = \frac{1}{2} \|Ax - y\|^2$$



Maximum A-Posteriori

Find x from observation $y = Ax + \xi$

Maximum A-Posteriori

$$x^* \in \arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

$$\iff \arg \min_{x \in \mathbb{R}^n} \text{data-fidelity} + \text{prior}$$

$$\text{e.g. } f(x) = \frac{1}{2} \|Ax - y\|^2$$



Plug & Play (PnP) [Venkatakrishnan et al., '13]

Find x from observation $y = Ax + \xi$

Maximum A-Posteriori

$$x^* \in \arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

✗ unknown prior

Plug & Play (PnP) [Venkatakrishnan et al., '13]

Minimize

$$\arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

Max

$$x^* \in$$

Plug & Play (PnP) [Venkatakrishnan et al., '13]

Minimize

$$\arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

Max

Proximal algorithms

$x^* \in$

- PGD: $x_{k+1} = \text{Prox}_{\tau g} \circ (\text{Id} - \tau \nabla f)(x_k)$
- DRS: $x_{k+1} = \frac{1}{2}x_k + \frac{1}{2}(2\text{Prox}_{\tau g} - \text{Id}) \circ (2\text{Prox}_{\tau f} - \text{Id})(x_k)$

Plug & Play (PnP) [Venkatakrishnan et al., '13]

Minimize

$$\arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

Max

Proximal algorithms

- PGD: $x_{k+1} = \text{Prox}_{\tau g} \circ (\text{Id} - \tau \nabla f)(x_k)$
- DRS: $x_{k+1} = \frac{1}{2}x_k + \frac{1}{2}(2\text{Prox}_{\tau g} - \text{Id}) \circ (2\text{Prox}_{\tau f} - \text{Id})(x_k)$

Proximal operator :

$$\text{Prox}_{\tau g}(y) = \arg \min_{x \in \mathbb{R}^n} \frac{1}{2\tau} \|x - y\|^2 + g(x)$$

Plug & Play (PnP) [Venkatakrishnan et al., '13]

Minimize

$$\arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

Max

Proximal algorithms

- PGD: $x_{k+1} = \text{Prox}_{\tau g} \circ (\text{Id} - \tau \nabla f)(x_k)$
- DRS: $x_{k+1} = \frac{1}{2}x_k + \frac{1}{2}(2\text{Prox}_{\tau g} - \text{Id}) \circ (2\text{Prox}_{\tau f} - \text{Id})(x_k)$

Proximal operator :

$$\text{Prox}_{\tau g}(y) = \arg \min_{x \in \mathbb{R}^n} \frac{1}{2\tau} \|x - y\|^2 + g(x)$$

\Leftrightarrow Gaussian Denoising MAP with prior g ($\sigma^2 = \tau$)

Plug & Play (PnP) [Venkatakrishnan et al., '13]

Minimize

$$\arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

Max

$x^* \in$

Proximal algorithms

- PGD: $x_{k+1} = \text{Prox}_{\tau g} \circ (\text{Id} - \tau \nabla f)(x_k)$
- DRS: $x_{k+1} = \frac{1}{2}x_k + \frac{1}{2}(2\text{Prox}_{\tau g} - \text{Id}) \circ (2 \text{Prox}_{\tau f} - \text{Id})(x_k)$

Proximal operator :

$$\text{Prox}_{\tau g}(y) = \arg \min_{x \in \mathbb{R}^n} \frac{1}{2\tau} \|x - y\|^2 + g(x) \approx \text{D}_{\sigma}(x)$$

\Leftrightarrow Gaussian Denoising MAP with prior g ($\sigma^2 = \tau$)

PnP algorithms

- PnP-PGD: $x_{k+1} = \text{D}_{\sigma} \circ (\text{Id} - \tau \nabla f)(x_k)$
- PnP-DRS: $x_{k+1} = \frac{1}{2}x_k + \frac{1}{2}(2\text{D}_{\sigma} - \text{Id}) \circ (2 \text{Prox}_{\tau f} - \text{Id})(x_k)$

Plug & Play (PnP) [Venkatakrishnan et al., '13]

Find x from observation $y = Ax + \xi$

Maximum A-Posteriori

$$x^* \in \arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

✗ unknown prior

$$D_\sigma \approx \text{Prox}_{\tau g}$$

Plug-and-Play (PnP)

✓ implicit prior

✓ $D_\sigma \approx \text{Prox}_{\tau g}$

Plug & Play (PnP) [Venkatakrishnan et al., '13]

Find x from observation $y = Ax + \xi$

Maximum A-Posteriori

$$x^* \in \arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

✗ unknown prior

$$D_\sigma \approx \text{Prox}_{\tau g}$$

Plug-and-Play (PnP)

- ✓ implicit prior
- ✓ SOTA restoration

✓ $D_\sigma \approx \text{Prox}_{\tau g}$

Plug & Play (PnP) [Venkatakrishnan et al., '13]

Find x from observation $y = Ax + \xi$

Maximum A-Posteriori

$$x^* \in \arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

✗ unknown prior

$$D_\sigma \approx \text{Prox}_{\tau g}$$

Plug-and-Play (PnP)

- ✓ implicit prior
- ✓ SOTA restoration
- ✗ no minimization problem
- ✗ no convergence guarantees

✗ $D_\sigma \neq \text{Prox}_{\tau g}$

Plug & Play (PnP) [Venkatakrishnan et al., '13]

Find x from observation $y = Ax + \xi$

How to ensure that D_σ is exactly a Prox ?

M

$$x^* \in \arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

✗ unknown prior

Plug-and-Play (PnP)

- ✓ implicit prior
- ✓ SOTA restoration
- ✗ no minimization problem
- ✗ no convergence guarantees

Play & Play (D_σD) [Moreau '65]

How to ensure that D_σ is exactly a Prox ?

[Moreau '65] If $D_\sigma = \nabla h_\sigma$ with :

- h_σ convex
- D_σ 1-Lipschitz

$$\Rightarrow D_\sigma = \text{Prox}_{\phi_\sigma} \text{ with } \phi_\sigma \text{ convex .}$$

- ✓ implicit prior
- ✓ SOTA restoration
- ✗ no minimization problem
- ✗ no convergence guarantees

Play & Play (D_σD) [Moreau '65, Rockafellar '70]

How to ensure that D_σ is exactly a Prox ?

[Moreau '65] If $D_\sigma = \nabla h_\sigma$ with :

- h_σ convex
- D_σ 1-Lipschitz \times non-realistic [Sun et al. '19, Bohra et al. '21]

$$\Rightarrow D_\sigma = \text{Prox}_{\phi_\sigma} \text{ with } \phi_\sigma \text{ convex .}$$

- ✓ implicit prior
- ✓ SOTA restoration
- ✗ no minimization problem
- ✗ no convergence guarantees

How to ensure that D_σ is exactly a Prox ?

[Gribonval & Nikolava '20] If $D_\sigma = \nabla h_\sigma$ with :

- h_σ convex
- D_σ 1-Lipschitz

$\Rightarrow D_\sigma = \text{Prox}_{\phi_\sigma}$ with ϕ_σ convex nonconvex.

- ✓ implicit prior
- ✓ SOTA restoration
- ✗ no minimization problem
- ✗ no convergence guarantees

Play & Play (DnD) [Kondor et al., 2019]

How to ensure that D_σ is exactly a Prox ?

[Gribonval & Nikolava '20] If $D_\sigma = \nabla h_\sigma$ with :

- h_σ convex
- $-D_\sigma$ 1-Lipschitz

$\Rightarrow D_\sigma = \text{Prox}_{\phi_\sigma}$ with ϕ_σ convex nonconvex.

✓ implicit prior

We train the denoiser $D_\sigma = \nabla h_\sigma$

with $h_\sigma : \mathbb{R}^n \rightarrow \mathbb{R}$ parametrized by a neural network.

Play & Play (Denoising)

How to ensure that D_σ is exactly a Prox ?

[Gribonval & Nikolava '20] If $D_\sigma = \nabla h_\sigma$ with :

- h_σ convex
- $-D_\sigma$ 1-Lipschitz

$\Rightarrow D_\sigma = \text{Prox}_{\phi_\sigma}$ with ϕ_σ convex nonconvex.

✓ implicit prior

We train the denoiser $D_\sigma = \nabla h_\sigma$

with $h_\sigma : \mathbb{R}^n \rightarrow \mathbb{R}$ parametrized by a neural network.

$$h_\sigma \text{ convexe} \Leftrightarrow \text{Id} - D_\sigma \text{ 1-Lipschitz} \Leftrightarrow \forall x, \|J_{(\text{Id} - D_\sigma)}(x)\|_S \leq 1$$

Training loss :

$$\mathbb{E}_{x, \xi_\sigma} \left[\|D_\sigma(x + \xi_\sigma) - x\|^2 + \mu \max(\|J_{(\text{Id} - D_\sigma)}(x + \xi_\sigma)\|_S, 1 - \epsilon) \right]$$

- ✓ D_σ achieves performant denoising.
- ✓ $\text{Id} - D_\sigma$ contractive.

PnP algorithms ...

- PnP-PGD: $x_{k+1} = \textcolor{red}{D}_\sigma \circ (\text{Id} - \tau \nabla f)(x_k)$
- PnP-DRS: $x_{k+1} = \frac{1}{2}x_k + \frac{1}{2}(2\textcolor{red}{D}_\sigma - \text{Id}) \circ (2 \text{Prox}_{\tau f} - \text{Id})(x_k)$

... (**with** $\tau = 1$) become again **proximal algorithms**

- Prox-PnP-PGD: $x_{k+1} = \text{Prox}_{\phi_\sigma} \circ (\text{Id} - \nabla f)(x_k)$
- Prox-PnP-DRS: $x_{k+1} = \frac{1}{2}x_k + \frac{1}{2}(2\text{Prox}_{\phi_\sigma} - \text{Id})(2 \text{Prox}_f - \text{Id})(x_k)$

• implicit prior

- ✓ SOTA restoration
- ✗ no minimization problem
- ✗ no convergence guarantees

PnP algorithms ...

- PnP-PGD: $x_{k+1} = \mathcal{D}_\sigma \circ (\text{Id} - \tau \nabla f)(x_k)$
- PnP-DRS: $x_{k+1} = \frac{1}{2}x_k + \frac{1}{2}(2\mathcal{D}_\sigma - \text{Id}) \circ (2 \text{Prox}_{\tau f} - \text{Id})(x_k)$

... (with $\tau = 1$) become again **proximal algorithms**

- Prox-PnP-PGD: $x_{k+1} = \text{Prox}_{\phi_\sigma} \circ (\text{Id} - \nabla f)(x_k)$
- Prox-PnP-DRS: $x_{k+1} = \frac{1}{2}x_k + \frac{1}{2}(2\text{Prox}_{\phi_\sigma} - \text{Id})(2 \text{Prox}_f - \text{Id})(x_k)$

Convergence analysis : nonconvex PGD and DRS

Objective function :

$$F = f + \phi_\sigma$$

[Attouch et al, '13] [Themelis & Patrinos, '20]

Under the right assumptions...

- $F(x_k)$ converges and $\|x_{k+1} - x_k\| \rightarrow 0$.
- (x_k) converges to a critical point of F .

Prox-PnP

Find x from observation $y = Ax + \xi$

Maximum A-Posteriori

$$x^* \in \arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

✗ unknown prior

Prox-PnP [Hurault et al, '22]

$$0 \in \partial(f + \phi_\sigma)(x^*)$$

- ✓ explicit prior
- ✓ minimization problem
- ✓ convergence guarantees

$$D_\sigma \approx \text{Prox}_{\tau g}$$

Plug-and-Play (PnP)

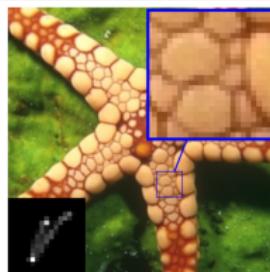
- ✓ implicit prior
- ✓ SOTA restoration
- ✗ no minimization problem
- ✗ no convergence guarantees

Prox-Denoiser
 $D_\sigma = \text{Prox}_{\phi_\sigma}$

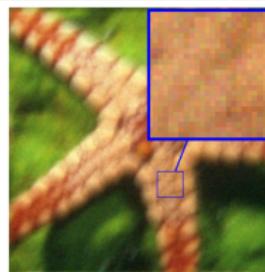


Prox-PnP

Deblurring with motion kernel and Gaussian noise std $\nu = 0.03$



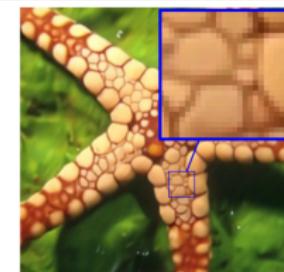
(a) Clean



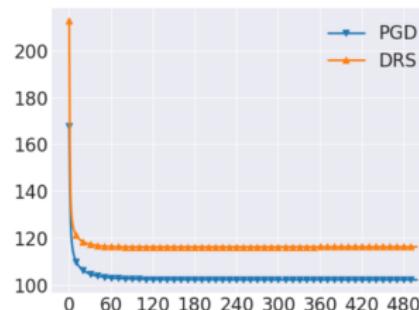
(b) Observed



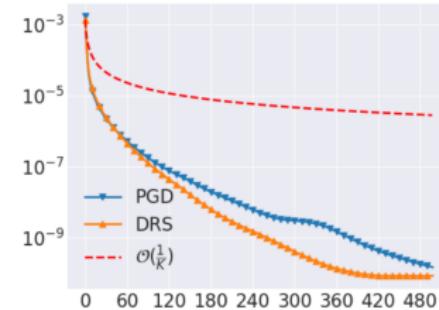
(c) Prox-PnP-PGD
(29.41dB)



(d) Prox-PnP-DRS
(29.65dB)



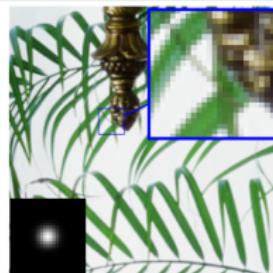
(e) $F_{\lambda,\sigma}(x_k)$



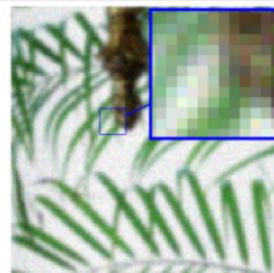
(f) $\min_{i \leq k} \|x_{i+1} - x_i\|^2$

Prox-PnP

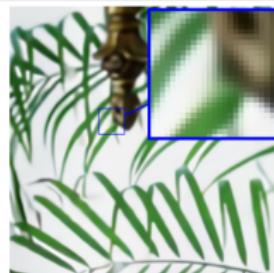
Super-resolution with Gaussian kernel and Gaussian noise std $\nu = 0.03$



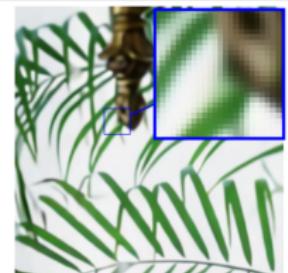
(a) Clean



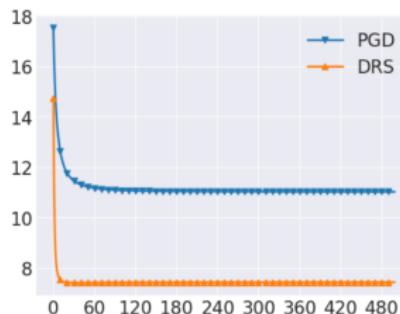
(b) Observed



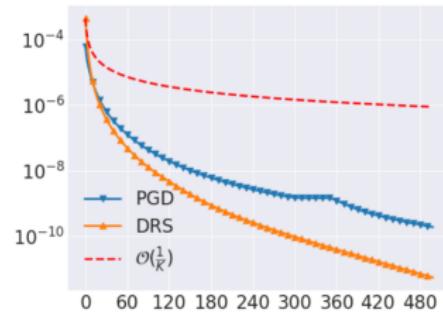
(c) Prox-PnP-PGD
(23.96dB)



(d) Prox-PnP-DRS
(24.36dB)



(e) $F_{\lambda, \sigma(x_k)}$



(f) $\min_{i \leq k} \|x_{i+1} - x_i\|^2$

Prox-PnP

Find x from observation $y = Ax + \xi$

Maximum A-Posteriori

$$x^* \in \arg \min_{x \in \mathbb{R}^n} f(x) + g(x)$$

✗ unknown prior

Prox-PnP [Hurault et al, '22]

$$0 \in \partial(f + \phi_\sigma)(x^*)$$

- ✓ explicit prior
- ✓ SOTA restoration
- ✓ minimization problem
- ✓ convergence guarantees

$$D_\sigma \approx \text{Prox}_{\tau g}$$

Plug-and-Play (PnP)

- ✓ implicit prior
- ✓ SOTA restoration
- ✗ no minimization problem
- ✗ no convergence guarantees

$$\text{Prox-Denoiser}$$
$$D_\sigma = \text{Prox}_{\phi_\sigma}$$



Thank you

Tonight 6 p.m. — 8 p.m. :
Poster #636 Hall E

