Plug-and-Play Image Restoration

Arthur Leclaire

Joint work with Samuel Hurault and Nicolas Papadakis

TELECOM Paris

IP PARIS

MVA Generative Modeling
February, 27th, 2024

Today

- We will introduce Plug-and-Play (PnP) methods to solve inverse problems
- We will give tools for **convergence analysis of PnP methods** based, today, on **fixed point theory** of ***averaged*** **operators**
- We will discuss the practical setup of such PnP algorithms

Plan

Plug-and-Play Algorithms

Convergence by Fixed Point Theory

PnP in practice

Image Inverse Problems

Find x_0 from observation $y = Ax_0 + \xi$

- $y \in \mathbb{R}^m$ observation
- $x_0 \in \mathbb{R}^n$ unknown input
- $A \in \mathbb{R}^{m \times n}$ degradation operator
- ξ random noise, often $\xi \sim \mathcal{N}(0, \nu^2 \mathrm{Id}_m)$

Image Inverse Problems

Find x_0 from observation $y = Ax_0 + \xi$

- $y \in \mathbb{R}^m$ observation
- $x_0 \in \mathbb{R}^n$ unknown input
- $A \in \mathbb{R}^{m \times n}$ degradation operator
- ξ random noise, often $\xi \sim \mathcal{N}(0, \nu^2 \mathrm{Id}_m)$

Denoising:


$=$

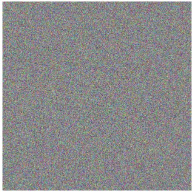
$+$


Image Inverse Problems

Find x_0 from observation $y = Ax_0 + \xi$

- $y \in \mathbb{R}^m$ observation
- $x_0 \in \mathbb{R}^n$ unknown input
- $A \in \mathbb{R}^{m \times n}$ degradation operator
- ξ random noise, often $\xi \sim \mathcal{N}(0, \nu^2 \mathrm{Id}_m)$

Deblurring:

Image Inverse Problems

Find x_0 from observation $y = Ax_0 + \xi$

- $y \in \mathbb{R}^m$ observation
- $x_0 \in \mathbb{R}^n$ unknown input
- $A \in \mathbb{R}^{m \times n}$ degradation operator
- ξ random noise, often $\xi \sim \mathcal{N}(0, \nu^2 \mathrm{Id}_m)$

Super-resolution:

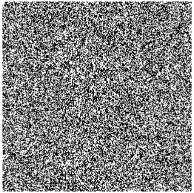
Image Inverse Problems

Find x_0 from observation $y = Ax_0 + \xi$

- $y \in \mathbb{R}^m$ observation
- $x_0 \in \mathbb{R}^n$ unknown input
- $A \in \mathbb{R}^{m \times n}$ degradation operator
- ξ random noise, often $\xi \sim \mathcal{N}(0, \nu^2 \mathrm{Id}_m)$

Inpainting:

$=$



$\otimes$

Image Inverse Problems

Find x_0 from observation $y = Ax_0 + \xi$

- $y \in \mathbb{R}^m$ observation
- $x_0 \in \mathbb{R}^n$ unknown input
- $A \in \mathbb{R}^{m \times n}$ degradation operator
- ξ random noise, often $\xi \sim \mathcal{N}(0, \nu^2 \mathrm{Id}_m)$

Compressed Sensing: *e.g.* Magnetic Resonance Imaging (MRI)

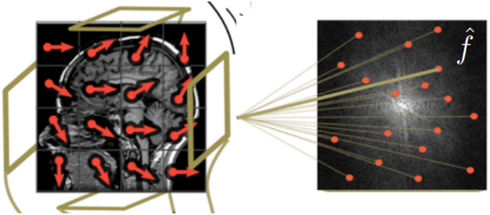


Image Inverse Problems

Find x_0 from observation $y \sim p(y|x_0)$

- $y \in \mathbb{R}^m$ observation
- $x_0 \in \mathbb{R}^n$ unknown input
- $p(y|x)$ forward model

Computed Tomography:

Maximum A-Posteriori

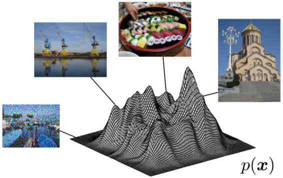
Find x from observation $y \sim p(y|x)$ with an a-priori $p(x)$ on the solution

$$\underset{x\in\mathbb{R}^n}{\operatorname{Argmax}}\, p(x|y) = \underset{x\in\mathbb{R}^n}{\operatorname{Argmax}} \frac{p(y|x)p(x)}{p(y)} = \underset{x\in\mathbb{R}^n}{\operatorname{Argmin}} -\log p(y|x) - \log p(x)$$

Maximum A-Posteriori

$$x^* \in \underset{x\in\mathbb{R}^n}{\operatorname{Argmin}}\ f(x) + \lambda\, g(x)$$

$$\iff \operatorname{Argmin}_{x\in\mathbb{R}^n} \underbrace{f(x) = -\log p(y|x)}_{\text{data-fidelity}} + \underbrace{g(x) \propto -\log p(x)}_{\text{regularization}}$$



A variety of data-fidelity terms f

- Assuming Gaussian noise model $\xi \sim \mathcal{N}(0, \nu^2 \mathrm{Id})$,

$$f(x) = -\log p(y|x) = \frac{1}{2\nu^2}||Ax - y||^2$$

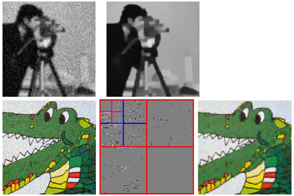
$\rightarrow$ *convex and smooth f, non-strongly convex in general*

- Less regular cases
    - Noiseless case: $f(x) = \imath_{\{x \mid Ax=y\}}$ $\rightarrow$ *non-smooth f*
    - Laplace / Poisson noise model $\rightarrow$ *non-smooth f*
    - Phase retrieval $\rightarrow$ *non-convex f*
- More complex non-linear modeling of real complex physical systems (e.g. X-ray computed tomography, electron-microscopy...)

A variety of explicit image priors

Design an explicit regularization on image features:

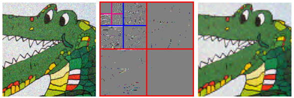
- Total variation (Rudin et al., 1992)
- Fourier spectrum (Ruderman, 1994)
- Wavelet sparsity (Mallat, 2009)



A variety of explicit image priors

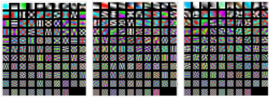
Design an explicit regularization on image features:

- Total variation (Rudin et al., 1992)
- Fourier spectrum (Ruderman, 1994)
- Wavelet sparsity (Mallat, 2009)



Learn an explicit prior on patches:

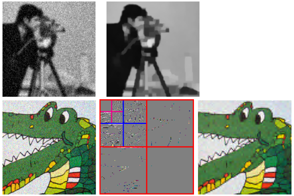
- Dictionary learning (Elad and Aharon, 2006), (Mairal et al., 2008)
- Gaussian mixture models (Yu et al., 2011), (Zoran and Weiss, 2011)



A variety of explicit image priors

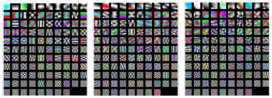
Design an explicit regularization on image features:

- Total variation (Rudin et al., 1992)
- Fourier spectrum (Ruderman, 1994)
- Wavelet sparsity (Mallat, 2009)



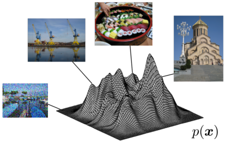
Learn an explicit prior on patches:

- Dictionary learning (Elad and Aharon, 2006), (Mairal et al., 2008)
- Gaussian mixture models (Yu et al., 2011), (Zoran and Weiss, 2011)



Learn an explicit deep prior on full images (generative models):

- Variational Auto-encoders (Kingma and Welling, 2019)
- Normalizing flows (Rezende and Mohamed, 2015)
- Score-based/Diffusion models (Song et al., 2021)



Plug-and-Play motivations

$$\text{Find } x^* \in \text{Argmin}_{x\in\mathbb{R}^n} \text{Data-fidelity}(x) + \text{Regularization}(x)$$

- **Decouple** data-fidelity and regularization via splitting algorithms (Combettes and Pesquet, 2011), (Zoran and Weiss, 2011)

✓ **Image Denoising** is relatively easy and well-understood.

→ State-of-the art denoisers **without explicit prior**

Filtering methods Dabov et al. (2007), Lebrun et al. (2013)

Deep denoisers Zhang et al. (2017b,a), Song et al. (2021)

→ Denoising is taking a step towards the manifold of clean images: **implicit prior**

We will alternate between

1. Taking a denoising step
2. Enforcing data-fidelity

First order optimization algorithms

Find $x^* \in \text{Argmin}_{x\in\mathbb{R}^n} F(x)$

- **Gradient Descent**

$$x_{k+1} = (\text{Id} - \tau\nabla F)(x_k) \quad \text{i.e.} \quad x_{k+1} = x_k - \tau\nabla F(x_k)$$

- **Proximal Point Algorithm**

$$x_{k+1} \in \text{Prox}_{\tau F}(x_k) \quad \text{i.e.} \quad x_{k+1} + \tau\nabla F(x_{k+1}) = x_k$$

$$\text{where} \quad \text{Prox}_F(y) := \underset{x\in\mathbb{R}^n}{\text{Argmin}} \frac{1}{2}||x-y||^2 + F(x)$$

Warning: Computing Prox_F (uniquely) requires some conditions on F, and is sometimes difficult.

Proximal Splitting

(Bauschke and Combettes, 2011)

Find $x^* \in \text{Argmin}_{x \in \mathbb{R}^n} f(x) + g(x)$

- Gradient Descent (GD)

$$x_{k+1} = (\text{Id} - \tau(\nabla f + \nabla g))(x_k)$$

- Proximal Gradient Descent (PGD, ISTA)

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

- Half Quadratic Splitting (HQS)

$$x_{k+1} = \text{Prox}_{\tau g} \circ \text{Prox}_{\tau f}(x_k)$$

does not target $f + g$

- Douglas-Rashford Splitting (DRS) / ADMM

$$x_{k+1} = \left(\frac{1}{2}\text{Id} + \frac{1}{2}(2\,\text{Prox}_{\tau g} - \text{Id}) \circ (2\,\text{Prox}_{\tau f} - \text{Id})\right)(x_k) \quad \text{and} \quad \tilde{x}_k = \text{Prox}_{\tau f}(x_k)$$

Denoising prior

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

- Input distribution $p(x)$.
- Gaussian noise $\xi \sim \mathcal{N}(0, \sigma^2 \mathrm{Id})$.
- Noisy observation y with density $p_\sigma(y)$ where $p_\sigma = p * \mathcal{N}(0, \sigma^2 \mathrm{Id})$.

MAP estimator

$$D_\sigma^{\mathrm{MAP}}(y) = \operatorname*{Argmax}_x p(x|y)$$

MMSE estimator

$$D_\sigma^{\mathrm{MMSE}}(y) = \mathbb{E}_{x \sim p(x|y)}[x]$$

Denoising prior

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

- Input distribution $p(x)$.
- Gaussian noise $\xi \sim \mathcal{N}(0, \sigma^2 \mathrm{Id})$.
- Noisy observation y with density $p_\sigma(y)$ where $p_\sigma = p * \mathcal{N}(0, \sigma^2 \mathrm{Id})$.

MAP estimator

$$D_\sigma^{\mathrm{MAP}}(y) = \operatorname*{Argmax}_x p(x|y)$$

MMSE estimator

$$D_\sigma^{\mathrm{MMSE}}(y) = \mathbb{E}_{x \sim p(x|y)}[x]$$

$$\begin{aligned}
\operatorname*{Argmax}_x p(x|y) &= \operatorname*{Argmax}_{x \in \mathbb{R}^n} \frac{p(y|x)p(x)}{p(y)} \\
&= \operatorname*{Argmin}_{x \in \mathbb{R}^n} -\log p(y|x) - \log p(x) \\
&= \operatorname*{Argmin}_{x \in \mathbb{R}^n} \frac{1}{2\sigma^2} \|x - y\|^2 - \log p(x) = \mathrm{Prox}_{-\sigma^2 \log p}(y)
\end{aligned}$$

Denoising prior

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

- Input distribution $p(x)$.
- Gaussian noise $\xi \sim \mathcal{N}(0, \sigma^2 \mathrm{Id})$.
- Noisy observation y with density $p_\sigma(y)$ where $p_\sigma = p * \mathcal{N}(0, \sigma^2 \mathrm{Id})$.

MAP estimator

$$D_\sigma^{\mathrm{MAP}}(y) = \operatorname*{Argmax}_x p(x|y)$$

MMSE estimator

$$D_\sigma^{\mathrm{MMSE}}(y) = \mathbb{E}_{x \sim p(x|y)}[x]$$

Bayes:

$$D_\sigma^{\mathrm{MAP}} = \mathrm{Prox}_{-\sigma^2 \log p}$$

Tweedie:

$$D_\sigma^{\mathrm{MMSE}} = \mathrm{Id} - \nabla(-\sigma^2 \log p_\sigma)$$

A denoiser is related to an **implicit prior**

PnP and RED algorithms

Find $x^* \in \operatorname{Argmin} f(x) + \lambda g(x)$ with $f = -\log p(y|.)$ and $g \propto -\log p$

$$
\begin{cases}
\text{GD} & : x_{k+1} = (\mathrm{Id} - \tau(\nabla f + \lambda \nabla g))(x_k) \\
\text{HQS} & : x_{k+1} = \mathrm{Prox}_{\tau\lambda g} \circ \mathrm{Prox}_{\tau f}(x_k) \\
\text{PGD} & : x_{k+1} = \mathrm{Prox}_{\tau\lambda g} \circ (\mathrm{Id} - \tau \nabla f)(x_k) \\
\text{DRS} & : x_{k+1} = \frac{1}{2}\mathrm{Id} + \frac{1}{2}(2\,\mathrm{Prox}_{\tau\lambda g} - \mathrm{Id}) \circ (2\,\mathrm{Prox}_{\tau f} - \mathrm{Id})(x_k)
\end{cases}
$$

PnP and RED algorithms

Find $x^* \in \text{Argmin}\, f(x) + \lambda g(x)$ with $f = -\log p(y|.)$ and $g \propto -\log p$

$$\begin{cases} \text{GD} & : x_{k+1} = (\text{Id} - \tau(\nabla f + \lambda \nabla g))(x_k) \\ \text{HQS} & : x_{k+1} = \text{Prox}_{\tau\lambda g} \circ \text{Prox}_{\tau f}(x_k) \\ \text{PGD} & : x_{k+1} = \text{Prox}_{\tau\lambda g} \circ (\text{Id} - \tau \nabla f)(x_k) \\ \text{DRS} & : x_{k+1} = \frac{1}{2}\text{Id} + \frac{1}{2}(2\,\text{Prox}_{\tau\lambda g} - \text{Id}) \circ (2\,\text{Prox}_{\tau f} - \text{Id})(x_k) \end{cases}$$

MAP denoiser

$$D_\sigma(y) = \text{Prox}_{-\sigma^2 \log p}(y)$$

MMSE denoiser

$$D_\sigma(y) = (\text{Id} + \sigma^2 \nabla \log p_\sigma)(y)$$

PnP and RED algorithms

Find $x^* \in \operatorname{Argmin} f(x) + \lambda g(x)$ with $f = -\log p(y|.)$ and $g \propto -\log p$

$$\begin{cases} \text{GD} & : x_{k+1} = (\text{Id} - \tau(\nabla f + \lambda \nabla g))(x_k) \\ \text{HQS} & : x_{k+1} = \text{Prox}_{\tau\lambda g} \circ \text{Prox}_{\tau f}(x_k) \\ \text{PGD} & : x_{k+1} = \text{Prox}_{\tau\lambda g} \circ (\text{Id} - \tau \nabla f)(x_k) \\ \text{DRS} & : x_{k+1} = \frac{1}{2}\text{Id} + \frac{1}{2}(2\,\text{Prox}_{\tau\lambda g} - \text{Id}) \circ (2\,\text{Prox}_{\tau f} - \text{Id})(x_k) \end{cases}$$

MAP denoiser

$$D_\sigma(y) = \text{Prox}_{-\sigma^2 \log p}(y)$$

MMSE denoiser

$$D_\sigma(y) = (\text{Id} + \sigma^2 \nabla \log p_\sigma)(y)$$

Plug-and-play (PnP)
(Venkatakrishnan et al., 2013)

$$\text{Prox}_{\tau\lambda g} \longrightarrow D_\sigma$$

Regularization by denoising (RED)
(Romano et al., 2017)

$$\text{Id} - \nabla g \longrightarrow D_\sigma$$

PnP and RED algorithms

$$\begin{cases} \text{GD} & : x_{k+1} = (\text{Id} - \tau(\nabla f + \lambda \nabla g))(x_k) \\ \text{HQS} & : x_{k+1} = \text{Prox}_{\tau\lambda g} \circ \text{Prox}_{\tau f}(x_k) \\ \text{PGD} & : x_{k+1} = \text{Prox}_{\tau\lambda g} \circ (\text{Id} - \tau \nabla f)(x_k) \\ \text{DRS} & : x_{k+1} = \frac{1}{2}\text{Id} + \frac{1}{2}(2\,\text{Prox}_{\tau\lambda g} - \text{Id}) \circ (2\,\text{Prox}_{\tau f} - \text{Id})(x_k) \end{cases}$$

MAP denoiser

$$D_\sigma(y) = \text{Prox}_{-\sigma^2 \log p}(y)$$

MMSE denoiser

$$D_\sigma(y) = (\text{Id} + \sigma^2 \nabla \log p_\sigma)(y)$$

PnP algorithms

(Venkatakrishnan et al., 2013)

$$\text{Prox}_{\tau\lambda g} \longrightarrow D_\sigma$$

RED algorithms

(Romano et al., 2017)

$$\text{Id} - \nabla g \longrightarrow D_\sigma$$

$$\begin{cases} \text{PnP-HQS} & : x_{k+1} = D_\sigma \circ \text{Prox}_{\tau f}(x_k) \\ \text{PnP-PGD} & : x_{k+1} = D_\sigma \circ (\text{Id} - \tau \nabla f)(x_k) \\ \text{PnP-DRS} & : x_{k+1} = \frac{1}{2}\text{Id} + \frac{1}{2}(2 D_\sigma - \text{Id}) \circ (2\,\text{Prox}_{\tau f} - \text{Id})(x_k) \end{cases}$$

PnP and RED algorithms

$$\begin{cases} \text{GD} & : x_{k+1} = (\text{Id} - \tau(\nabla f + \lambda \nabla g))(x_k) \\ \text{HQS} & : x_{k+1} = \text{Prox}_{\tau\lambda g} \circ \text{Prox}_{\tau f}(x_k) \\ \text{PGD} & : x_{k+1} = \text{Prox}_{\tau\lambda f} \circ (\text{Id} - \tau \nabla g)(x_k) \\ \text{DRS} & : x_{k+1} = \frac{1}{2}\text{Id} + \frac{1}{2}(2\,\text{Prox}_{\tau\lambda g} - \text{Id}) \circ (2\,\text{Prox}_{\tau f} - \text{Id})(x_k) \end{cases}$$

MAP denoiser

$$D_\sigma(y) = \text{Prox}_{-\sigma^2 \log p}(y)$$

MMSE denoiser

$$D_\sigma(y) = (\text{Id} + \sigma^2 \nabla \log p_\sigma)(y)$$

PnP algorithms

(Venkatakrishnan et al., 2013)

$$\text{Prox}_{\tau\lambda g} \longrightarrow D_\sigma$$

RED algorithms

(Romano et al., 2017)

$$\text{Id} - \nabla g \longrightarrow D_\sigma$$

$$\begin{cases} \text{RED-GD} & : x_{k+1} = (\tau\lambda D_\sigma + (1-\tau\lambda)\text{Id} - \tau\nabla f)(x_k) \\ \text{RED-PGD} & : x_{k+1} = \text{Prox}_{\tau f} \circ (\tau\lambda D_\sigma + (1-\tau\lambda)\text{Id})(x_k) \end{cases}$$

What about convergence?

⚠ But in practice, $D_\sigma \neq \text{Prox}_{\tau g}$, $D_\sigma \neq \text{Id} - \nabla g$...

Goal: Find minimal conditions on D_σ to get back convergence guarantees.

Plan

Plug-and-Play Algorithms

Convergence by Fixed Point Theory

PnP in practice

PnP Convergence by Fixed Point

The previous PnP algorithms can be written as

$$x_{k+1} = T_{PnP}(x_k)$$

$$\text{with} \quad T_{PnP} = \begin{cases} T_{HQS} & = D_\sigma \circ \operatorname{Prox}_{\tau f} \\ T_{PGD} & = D_\sigma \circ (\operatorname{Id} - \tau \nabla f) \\ T_{DRS} & = \frac{1}{2}\operatorname{Id} + \frac{1}{2}(2D_\sigma - \operatorname{Id}) \circ (2\operatorname{Prox}_{\tau f} - \operatorname{Id}) \end{cases}$$

Goal: Show that $x_k \to x^* \in \operatorname{Fix}(T_{PnP})$.

Averaged operator theory

(Bauschke and Combettes, 2011)

Let $T : \mathbf{R}^n \to \mathbf{R}^n$. We will consider $\mathbf{R}^n$ equipped with **the Euclidean norm**.

Definition

We say that T is nonexpansive if it is 1-Lipschitz.

Definition

T is θ**-averaged** (with $\theta \in (0,1)$) if there exists a nonexpansive $R : \mathbf{R}^n \to \mathbf{R}^n$ such that

$$T = \theta R + (1-\theta)\mathrm{Id}.$$

- "T θ-averaged" is equivalent to "$(1-\frac{1}{\theta})\mathrm{Id} + \frac{1}{\theta} T$ nonexpansive", and also to

$$\forall x, y \in \mathbf{R}^n,\ \|T(x) - T(y)\|^2 + \frac{1-\theta}{\theta}\|(\mathrm{Id}-T)(x) - (\mathrm{Id}-T)(y)\|^2 \leq \|x-y\|^2.$$

- T is θ-averaged $\implies$ T is nonexpansive.
- T is $\frac{1}{2}$-averaged $\iff$ T is firmly nonexpansive.

Composition of Averaged operators

Proposition

Let T be θ-averaged and $\alpha \in [0,1]$. Then

- $\alpha T + (1-\alpha)\mathrm{Id}$ *is* $\alpha\theta$*-averaged.*
- T *is* θ'*-averaged for any* $\theta' \in [\theta, 1]$.

Remark: If T is L-Lipschitz with $L < 1$, then T is $\frac{L+1}{2}$-averaged.

Proposition (Combettes and Yamada, 2015)

Let T_1 be θ_1-averaged and T_2 be θ_2-averaged, with any $\theta_1, \theta_2 \in (0,1)$.
Then $T_1 \circ T_2$ is θ-averaged with $\theta = \frac{\theta_1+\theta_2-2\theta_1\theta_2}{1-\theta_1\theta_2} \in (0,1)$.

Fixed Point Theorem for Averaged Operators

Theorem (Krasnosel'skiĭ-Mann)

Let $T : \mathbf{R}^n \to \mathbf{R}^n$ be a θ-averaged operator such that $\mathrm{Fix}(T) \neq \varnothing$.
Then the sequence $x_{k+1} = T(x_k)$ converges to a fixed point of T.

Sketch of proof (See (Bauschke and Combettes, 2011) or C. Dossal's lecture notes).

Write $T = \theta R + (1-\theta)\mathrm{Id}$ with R 1-Lipschitz and $\mathrm{Fix}(R) = \mathrm{Fix}(T)$. For $y \in \mathrm{Fix}(T)$,

- $\|x_{n+1} - y\|^2 \leqslant \|x_n - y\|^2 - \theta(1-\theta)\|Rx_n - x_n\|^2$
- $\sum_{n\in\mathbb{N}} \theta(1-\theta)\|Rx_n - x_n\|^2 \leqslant \|x_0 - y\|^2$
- $\|Rx_{n+1} - x_{n+1}\| = \|Rx_{n+1} - Rx_n + (1-\theta)(Rx_n - x_n)\| \leqslant \|Rx_n - x_n\|$.
- Since $(\|x_n - y\|)$ is non-increasing, there is a converging subsequence $x_{n_k} \to x$
- $Rx_n - x_n \to 0$ and thus $Rx_{n_k} \to x$, and also to Tx, thus $Rx = x$.
- Taking $y = x$, we get that $\|x_n - x\|$ is non-increasing with a subsequence converging to 0.

□

Remark: The theorem does not apply to $T = -\mathrm{Id}$ of course...

Proximity operator of Convex Functions

Let $\Gamma_0(\mathbf{R}^n)$ be the set of $f : \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$ that are convex, l.s.c., and proper (i.e. $f \not\equiv +\infty$).

For $f : \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$, and $x \in \mathbf{R}^n$, we define the subgradient

$$\partial f(x) = \{\ v \in \mathbf{R}^n \mid \forall z \in \mathbf{R}^n,\ \ f(z) \geqslant f(x) + \langle v, z - x \rangle\ \}.$$

It is easy to see that the subgradient of a proper function f is monotone, that is,

$$\forall x_1, x_2 \in \mathbf{R}^n, \forall v_1 \in \partial f(x_1), \forall v_2 \in \partial f(x_2), \quad \langle v_1 - v_2, x_1 - x_2 \rangle \geqslant 0.$$

Proposition

For $f \in \Gamma_0(\mathbf{R}^n)$, for any $x \in \mathbf{R}^n$, we can uniquely define

$$\mathrm{Prox}_f(x) = \operatorname*{Argmin}_{z \in \mathbf{R}^n} f(z) + \frac{1}{2}\|z - x\|^2.$$

The point $p = \mathrm{Prox}_f(x)$ is characterized by $x - p \in \partial f(p)$.

Consequence: If $f \in \Gamma_0(\mathbf{R}^n)$, then Prox_f is $\frac{1}{2}$-averaged (i.e. firmly nonexpansive).

Gradient-step of Convex Functions

Proposition

If $f : \mathbb{R}^n \to \mathbb{R}$ is convex, differentiable with L-Lipschitz gradient.
Then, for $\tau \in (0, \frac{2}{L})$, $\mathrm{Id} - \tau \nabla f$ is $\frac{\tau L}{2}$-averaged.

The proof relies on the observation that $\mathrm{Id} - \frac{2}{L}\nabla f$ is 1-Lipschitz, which is equivalent to

$$\forall x, z \in \mathbf{R}^n, \quad \frac{1}{L}\|\nabla f(x) - \nabla f(z)\|^2 \leqslant \langle \nabla f(x) - \nabla f(z), x - z \rangle.$$

(We sometimes say that ∇f is $\frac{1}{L}$-co-coercive, which is equivalent to $\frac{1}{L}\nabla f$ firmly nonexpansive.)

Consequence: Convergence of gradient descent for convex functions if there is as solution.

Remark: Under the same hypotheses, we can show that $\mathrm{Prox}_{\tau f}$ is $\frac{\tau L}{2(1+\tau L)}$-averaged for any $\tau > 0$.

Argmin and Fixed Points

Proposition

Let $f, g : \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$ *be proper l.s.c. with* f *differentiable, and let* $\tau > 0$. *Then*

$$\mathrm{Argmin}(f + g) = \mathrm{Fix}\left(\mathrm{Prox}_{\tau g} \circ (\mathrm{Id} - \tau \nabla f)\right).$$

Argmin and Fixed Points

Proposition

Let $f, g : \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$ be proper l.s.c. with f differentiable, and let $\tau > 0$. Then

$$\operatorname{Argmin}(f+g) = \operatorname{Fix}\left(\operatorname{Prox}_{\tau g} \circ (\operatorname{Id} - \tau \nabla f)\right).$$

Proof.

$$\begin{aligned} x \in \operatorname{Argmin}(f+g) &\iff 0 \in \nabla f(x) + \partial g(x) \\ &\iff -\tau \nabla f(x) \in \partial \tau g(x) \\ &\iff x \in \operatorname{Prox}_{\tau g}(x - \tau \nabla f(x)). \end{aligned}$$

□

In order to minimize $f+g$, it is thus relevant to study the convergence of the iterative sequence

$$x_{k+1} = \operatorname{Prox}_{\tau g}(x_k - \tau \nabla f(x_k)).$$

Reflected Proxity Operator

We define

$$\mathrm{RProx}_f = 2\,\mathrm{Prox}_f - \mathrm{Id}.$$

Then, Prox_f is $\frac{1}{2}$-averaged if and only if RProx_f is 1-Lipschitz.

Proposition

Let $f, g \in \Gamma_0(\mathbf{R}^n)$ *and let* $\tau > 0$. *Then*

$$\mathrm{Argmin}(f+g) = \mathrm{Prox}_{\tau f}\left(\mathrm{Fix}\left(\mathrm{RProx}_{\tau g} \circ \mathrm{RProx}_{\tau f}\right)\right).$$

Reflected Proxity Operator

We define

$$\mathrm{RProx}_f = 2\,\mathrm{Prox}_f - \mathrm{Id}.$$

Then, Prox_f is $\frac{1}{2}$-averaged if and only if RProx_f is 1-Lipschitz.

Proposition

Let $f, g \in \Gamma_0(\mathbf{R}^n)$ *and let* $\tau > 0$. *Then*

$$\mathrm{Argmin}(f+g) = \mathrm{Prox}_{\tau f}\Big(\mathrm{Fix}\left(\mathrm{RProx}_{\tau g} \circ \mathrm{RProx}_{\tau f}\right)\Big).$$

If $f, g \in \Gamma_0(\mathbf{R}^n)$, then $\frac{1}{2}\mathrm{Id} + \frac{1}{2}\mathrm{RProx}_{\tau g} \circ \mathrm{RProx}_{\tau f}$ is $\frac{1}{2}$-averaged.

Reflected Proxity Operator

We define

$$\mathrm{RProx}_f = 2\,\mathrm{Prox}_f - \mathrm{Id}.$$

Then, Prox_f is $\frac{1}{2}$-averaged if and only if RProx_f is 1-Lipschitz.

Proposition

Let $f, g \in \Gamma_0(\mathbf{R}^n)$ and let $\tau > 0$. Then

$$\mathrm{Argmin}(f+g) = \mathrm{Prox}_{\tau f}\left(\mathrm{Fix}\left(\mathrm{RProx}_{\tau g} \circ \mathrm{RProx}_{\tau f}\right)\right).$$

If $f, g \in \Gamma_0(\mathbf{R}^n)$, then $\frac{1}{2}\mathrm{Id} + \frac{1}{2}\,\mathrm{RProx}_{\tau g} \circ \mathrm{RProx}_{\tau f}$ is $\frac{1}{2}$-averaged.

In order to minimize $f + g$, it is thus relevant to study the convergence of the iterative sequence

$$x_{k+1} = \left(\frac{1}{2}\mathrm{Id} + \frac{1}{2}\,\mathrm{RProx}_{\tau g} \circ \mathrm{RProx}_{\tau f}\right)(x_k) \quad \text{and set} \quad \tilde{x}_k = \mathrm{Prox}_{\tau f}(x_k).$$

Averaged operator theory for PnP convergence

PnP algorithms:

$$x_{k+1} = T_{PnP}(x_k)$$

$$\text{with} \quad T_{PnP} = \begin{cases} T_{HQS} & = D_\sigma \circ \text{Prox}_{\tau f} \\ T_{PGD} & = D_\sigma \circ (\text{Id} - \tau \nabla f) \\ T_{DRS} & = \frac{1}{2}\text{Id} + \frac{1}{2}(2D_\sigma - \text{Id}) \circ (2\,\text{Prox}_{\tau f} - \text{Id}) \end{cases}$$

Theorem

*Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex, differentiable with ∇f L-Lipschitz, and D_σ be θ-averaged, $\theta \in (0, 1)$. Assume that the iterated operator **has** a fixed point.*

- ***PnP-HQS** converges towards a fixed point of T_{HQS}.*
- *If $\tau L < 2$, **PnP-PGD** converges towards a fixed point of T_{PGD}.*
- *If $\theta \leq 1/2$, **PnP-DRS** converges towards a fixed point of T_{DRS}.*

Averaged operator theory for PnP convergence

PnP algorithms:

$$x_{k+1} = T_{PnP}(x_k)$$

$$\text{with} \quad T_{PnP} = \begin{cases} T_{HQS} & = D_\sigma \circ \mathrm{Prox}_{\tau f} \\ T_{PGD} & = D_\sigma \circ (\mathrm{Id} - \tau \nabla f) \\ T_{DRS} & = \frac{1}{2}\mathrm{Id} + \frac{1}{2}(2D_\sigma - \mathrm{Id}) \circ (2\,\mathrm{Prox}_{\tau f} - \mathrm{Id}) \end{cases}$$

Theorem

Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex, differentiable with ∇f L-Lipschitz, and D_σ be θ-averaged, $\theta \in (0, 1)$. Assume that the iterated operator ***has*** *a fixed point.*

- ***PnP-HQS*** *converges towards a fixed point of T_{HQS}.*
- *If $\tau L < 2$,* ***PnP-PGD*** *converges towards a fixed point of T_{PGD}.*
- *If $\theta \leq 1/2$,* ***PnP-DRS*** *converges towards a fixed point of T_{DRS}.*

Remark:

✗ Does not extend to **nonconvex** data-fidelity terms f.

- If f is L-smooth and **strongly convex**, for $\tau L < 2$, $\mathrm{Id} - \tau \nabla f$ is contractive (Ryu et al., 2019)
  $\rightarrow$ allows to relax the denoiser hypothesis for D_σ $(1 + \epsilon)$-Lipschitz.

Plan

Plug-and-Play Algorithms

Convergence by Fixed Point Theory

PnP in practice

Which denoiser to use?

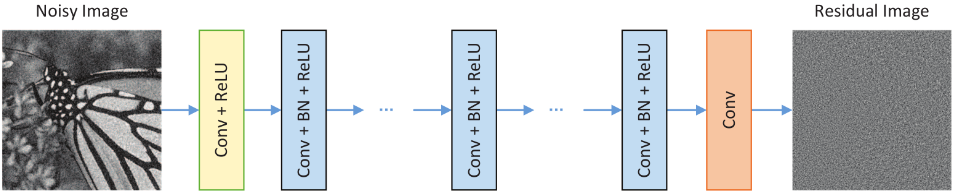
- One can use off-the-shelf denoisers: BM3D (Dabov et al., 2007), NLBayes (Lebrun et al., 2013), ...
- One can also use denoisers given as neural networks.
- Such a deep denoiser is trained to approximate the MMSE:

$$\underset{\text{Param}(D_\sigma)}{\text{Argmin}} \; \mathbb{E}_{x\sim p_X,\xi\sim\mathcal{N}(0,\sigma^2\text{Id})}\left[\|D_\sigma(x+\xi)-x\|^2\right]$$

  where p_X is a data distribution of clean images.
- For training, L^1 loss (instead of squared L^2 loss) sometimes gives better results.
- For certain denoising architectures, the noise level σ is given as input.
  ⚠ In PnP, the denoising strength σ may be different from the noise level of the input!

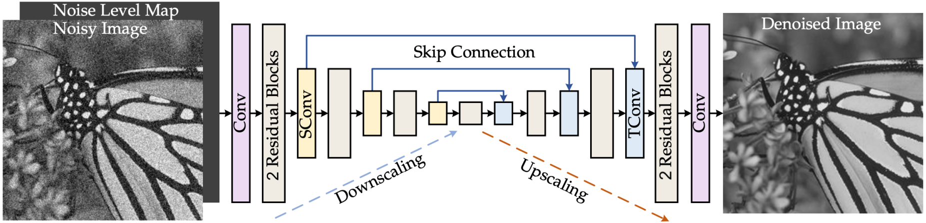
DnCNN (Zhang et al., 2017a)

- DnCNN is a deep convolutional neural network for denoising (Zhang et al., 2017a).
- It is based on residual learning: $D(x) = x + R(x)$ where R is the network.
- R has 20 layers of hidden dimension 64 (3×3 convolutions, BatchNorm, ReLU)
- It is trained on noise levels $\sigma \in [0, 50]$ and can be applied blindly (without σ).



DRUNet (Zhang et al., 2021)

- DRUNet is a deep convolutional neural network for denoising (Zhang et al., 2021).
- It is a UNet that includes residual blocks, convolutions (bias-free!), and skip connections.
- The UNet has 4 “scales” of dimensions $64, 128, 256, 512$.
- It is trained on noise levels $\sigma \in [0, 50]$ and take a noise level map as input.
- Zhang et al. (2021) propose to do PnP image restoration with this denoising prior (DPIR).



How to build averaged deep denoisers ?

How to build averaged deep denoisers ?

- Weight clipping or Spectral normalization (Miyato et al., 2018), (Ryu et al., 2019)
  ✗ Lipschitz constant $\ll 1$ for large networks
- Or we can penalize a Lipschitz constant in the training loss:

$$\underset{\text{Param}(D_\sigma)}{\text{Argmin}}\ \mathbb{E}_{x\sim p_X,\xi\sim\mathcal{N}(0,\sigma^2\text{Id})}\left[\|D_\sigma(x+\xi)-x\|^2\right]+\mu\text{Lip}(D_\sigma).$$

- Convolutional Proximal Neural Networks (Hertrich et al., 2021)
- Firmly nonexpansive denoisers (Terris et al., 2020)
- Deep spline neural networks (Goujon et al., 2023)
- $D_\sigma = \text{Id} - \nabla g_\sigma$ with g_σ Input Convex Neural Network (ICNN) (Meunier et al., 2022)

Non-expansiveness can harm denoising performance.

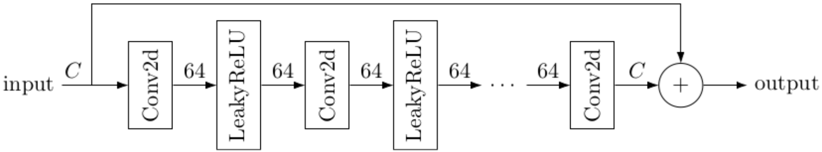
Nonexpansive convolutional neural networks (Pesquet et al., 2021)

Idea:

- Build a nonexpansive convolutional neural network (CNN)

$$D = T_M \circ \cdots \circ T_1 \quad \text{with} \quad T_m(x) = R_m(W_m x + b_m)$$

  where R_m is an (averaged) activation function, W_m a convolution, and b_m a bias.
- We want to have $D = \frac{\mathrm{Id}+Q}{2}$ with Q nonexpansive.
- During training, the Lipschitz constant of $2D - \mathrm{Id}$ is penalized.



Convolution proximal neural networks (Hertrich et al., 2021)

Idea: Build a convolutional proximal neural network (cPNN)

$$\Phi_u = T_M \circ \cdots \circ T_1 \quad \text{with} \quad T_m(x) = W_m^T \sigma_m(W_m x + b_m)$$

where $u = (W_m, \sigma_m, b_m)_{1 \leq m \leq M}$ is a collection of parameters.
The linear operators W_m (or W_m^T) are convolutions lying in a Stiefel manifold

$$\mathrm{St}(d, n) = \{ \, W \in \mathbf{R}^{n \times d} \mid W^T W = \mathrm{Id} \, \}.$$

The resulting denoiser is then $D = \mathrm{Id} - \gamma \Phi_u$.

- Ideally, Φ_u is a composition of M firmly non-expansive operators, thus averaged.
- In practice, W_m is a convolution with limited filter length.
- Condition $W_m \in \mathrm{St}$ is approximated with a term $\|W_m^T W_m - \mathrm{Id}\|_F^2$ in the learning cost.
- Φ_u is verified in practice to be t-averaged with t close to $\frac{1}{2}$.

Deep Spline Neural Networks (Goujon et al., 2023)

Idea: Approximate the proximal operator of a convex-ridge regularizer

$$R(x) = \sum_{p=1}^{P} \sum_{i} \psi_p(h_p * x(i))$$

where h_p are convolution kernels, and ψ_p are particular $\mathcal{C}^1$ convex functions.

Given a noisy z,

$$\text{Prox}_{\lambda R}(z) = \underset{x \in \mathbf{R}^n}{\text{Argmin}} \frac{1}{2}\|x - z\|^2 + \lambda R(x)$$

is approximated with t iterations of the gradient-step

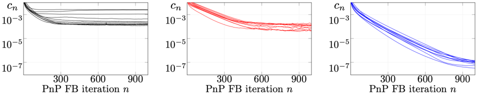
$$x \mapsto x - \alpha((x - z) + \lambda \nabla R(x)).$$

The output after t iterations is denoted by $T^t_{R,\lambda,\alpha}(z)$.

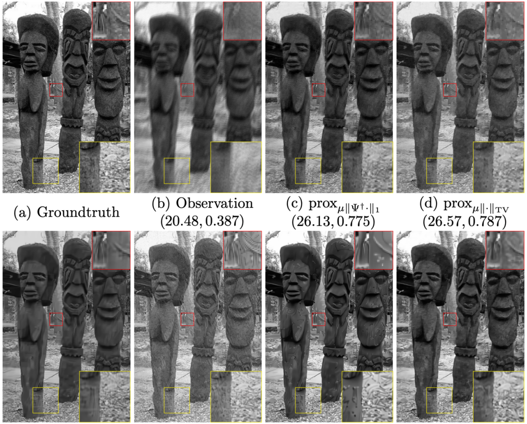
- $T^t_{R,\lambda,\alpha}$ approximates the prox of a convex function
- Linear spline parameterization of ψ_p justified by a density result

Example for Image Deblurring (Pesquet et al., 2021)

- PnP-PGD (aka Forward-Backward)
- Denoiser: Adapted DnCNN
- Below, evolution of $c_n = \frac{\|x_n - x_{n-1}\|}{\|x_0\|}$



(a) BM3D (b) RealSN (c) Proposed



(a) Groundtruth (b) Observation (20.48, 0.387) (c) $\mathrm{prox}_{\mu\|\Psi^\dagger\cdot\|_1}$ (26.13, 0.775) (d) $\mathrm{prox}_{\mu\|\cdot\|_{\mathrm{TV}}}$ (26.57, 0.787)

(e) BM3D (26.09, 0.732) (f) RealSN (24.68, 0.726) (g) DnCNN (26.12, 0.643) (h) Proposed (**27.09**, **0.789**)

Take-home Messages

- Convergence by fixed point relies on a particular kind of non-expansiveness.
- For now, we cannot enforce non-expansiveness exactly in practice.
  Instead we penalize some Lipschitz constant when training the denoiser.
- In that way, PnP methods lead to very good restoration results.
- Once learned a denoiser, it can be used to address many other inverse problems.
- PnP algorithms are (surprisingly) stable as soon as parameters are properly adjusted.
- Numerical control can be improved by relying on explicit minimization (see next week)
- Visual results can be further improved by tuning the strategy on σ ($\to$ diffusion models)

Et avant le TP, une petite page de publicité...

- S. Hurault's thesis on PnP algorithms: `https://www.theses.fr/2023BORD0336`
- A nice document on gradient descent by Robert Gower:
  `https://perso.telecom-paristech.fr/rgower/pdf/M2_statistique_optimisation/grad_conv.pdf`
  See also the handbook (Garrigos and Gower, 2023) or C. Dossal's lecture notes.
- Imaging in Paris seminar: `https://imaging-in-paris.github.io/`
- M2 internship on PnP methods for Hyperspectral Unmixing (with C. Kervazo and yours truly)
- Python/Pytorch library for Plug-and-Play Imaging:

Deep Inverse

`https://deepinv.github.io/`

Main contributors: S. Hurault, J. Tachella, M. Terris

THANK YOU FOR YOUR ATTENTION!

References I

Bauschke, H. H. and Combettes, P. L. (2011). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer.

Combettes, P. L. and Pesquet, J.-C. (2011). Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer.

Combettes, P. L. and Yamada, I. (2015). Compositions and convex combinations of averaged nonexpansive operators. *Journal of Mathematical Analysis and Applications*, 425(1):55–70.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. O. (2007). Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Processing*, 16(8):2080–2095.

Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745.

Garrigos, G. and Gower, R. M. (2023). Handbook of convergence theorems for (stochastic) gradient methods.

Goujon, A., Neumayer, S., Bohra, P., Ducotterd, S., and Unser, M. (2023). A neural-network-based convex regularizer for inverse problems. *IEEE Transactions on Computational Imaging*.

Hertrich, J., Neumayer, S., and Steidl, G. (2021). Convolutional proximal neural networks and plug-and-play algorithms. *Linear Algebra and its Applications*, 631:203–234.

References II

Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*.

Lebrun, M., Buades, A., and Morel, J. (2013). A nonlocal Bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688.

Mairal, J., Elad, M., and Sapiro, G. (2008). Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1).

Mallat, S. (2009). *A Wavelet Tour of Signal Processing, The Sparse Way*. Academic Press, Elsevier, 3rd edition edition.

Meunier, L., Delattre, B. J., Araujo, A., and Allauzen, A. (2022). A dynamical system perspective for Lipschitz neural networks. In *International Conference on Machine Learning*, pages 15484–15500. PMLR.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral Normalization for Generative Adversarial Networks. In *Proceedings of the 6th International Conference on Learning Representations, ICLR*.

Pesquet, J.-C., Repetti, A., Terris, M., and Wiaux, Y. (2021). Learning maximally monotone operators for image recovery. *SIAM Journal on Imaging Sciences*, 14(3):1206–1237.

References III

Rezende, D. J. and Mohamed, S. (2015). Variational inference with normalizing flows. In Bach, F. R. and Blei, D. M., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1530–1538. JMLR.org.

Romano, Y., Elad, M., and Milanfar, P. (2017). The little engine that could: Regularization by denoising (RED). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844.

Ruderman, D. L. (1994). The statistics of natural images. *Network: computation in neural systems*, 5(4):517.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268.

Ryu, E., Liu, J., Wang, S., Chen, X., Wang, Z., and Yin, W. (2019). Plug-and-play methods provably converge with properly trained denoisers. In *International Conference on Machine Learning*, pages 5546–5557. PMLR.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net.

References IV

Terris, M., Repetti, A., Pesquet, J.-C., and Wiaux, Y. (2020). Building firmly nonexpansive convolutional neural networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8658–8662. IEEE.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. (2013). Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE.

Yu, G., Sapiro, G., and Mallat, S. (2011). Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. *IEEE Transactions on Image Processing*, 21(5):2481–2499.

Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., and Timofte, R. (2021). Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017a). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155.

Zhang, K., Zuo, W., Gu, S., and Zhang, L. (2017b). Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938.

References V

Zoran, D. and Weiss, Y. (2011). From learning models of natural image patches to whole image restoration. In *2011 International Conference on Computer Vision*, pages 479–486. IEEE.