

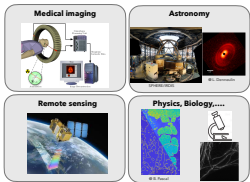
Multiresolution in image restoration

1- Introduction

Elisa Riccietti, Nelly Pustelnik

Notations and basics

Forward model: quantities of interest



→ Variables of interest

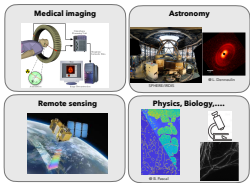
- $z \in \mathbb{R}^M$: data/measurements.
- $\bar{x} \in \mathbb{R}^N$: unknown image.
- $\hat{x} \in \mathbb{R}^N$: estimated image.

→ Forward model:

$$z = \mathcal{D}(\mathcal{A}\bar{x})$$

Stochastic degradation **Linear** operator

Forward model: quantities of interest



→ Variables of interest

- $z \in \mathbb{R}^M$: data/measurements.
- $\bar{x} \in \mathbb{R}^N$: unknown image.
- $\hat{x} \in \mathbb{R}^N$: estimated image.

→ Forward model:

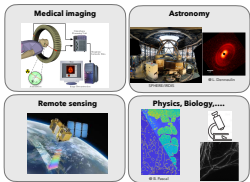
$$z = \mathcal{D}(\mathbf{A}\bar{x})$$

Stochastic degradation Linear operator

→ Inverse model:

$$\hat{x} = \mathbf{d}_{\Theta}(z)$$

Forward model: quantities of interest



→ Variables of interest

- $z \in \mathbb{R}^M$: data/measurements.
- $\bar{x} \in \mathbb{R}^N$: unknown image.
- $\hat{x} \in \mathbb{R}^N$: estimated image.

→ Forward model:

$$z = \mathcal{D}(\mathbf{A}\bar{x})$$

Stochastic degradation Linear operator

→ Inverse model:

$$\hat{x} = \mathbf{d}_{\Theta}(z)$$

→ **Goal:** Estimate $\hat{x}$ close to $\bar{x}$ from the information included in z , from the full or partial knowledge of $\mathbf{A}$, from noise statistics $\mathcal{D}$, and from *a priori* knowledge.

Vector representation of an image

Image = matrix of pixels $x \in \mathbb{R}^{N_1 \times N_2}$



255	255	253	252	217
255	236	227	241	182
250	172	187	170	110
238	166	90	103	82
235	158	79	81	56



In what follows:

image = vector $x \in \mathbb{R}^N$ with $N = N_1 N_2$

$$z = A\bar{x} \quad \Leftrightarrow \quad z = \phi * \bar{x}$$

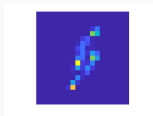
Basics about A: blur

$$z = A\bar{x} \quad \Leftrightarrow \quad z = \phi * \bar{x}$$



z

=



ϕ

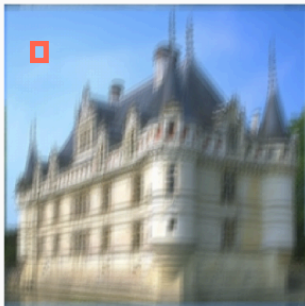
*



$\bar{x}$

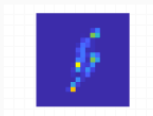
Basics about A: blur

$$z = A\bar{x} \quad \Leftrightarrow \quad z = \phi * \bar{x}$$



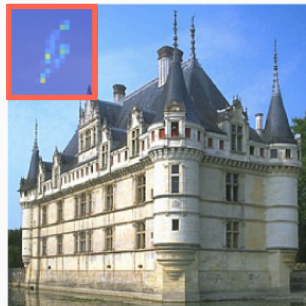
z

=



ϕ

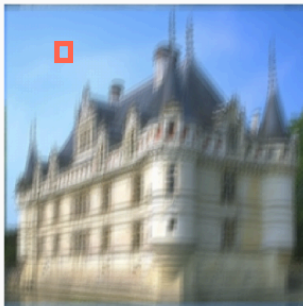
*



$\bar{x}$

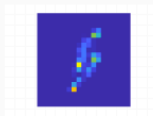
Basics about A: blur

$$z = A\bar{x} \quad \Leftrightarrow \quad z = \phi * \bar{x}$$



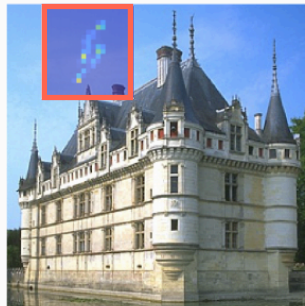
z

=



ϕ

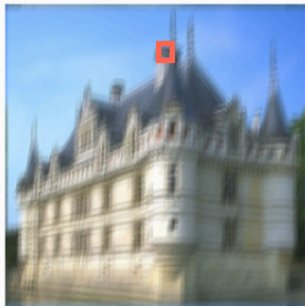
*



$\bar{x}$

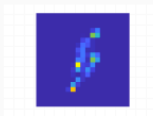
Basics about A: blur

$$z = A\bar{x} \quad \Leftrightarrow \quad z = \phi * \bar{x}$$



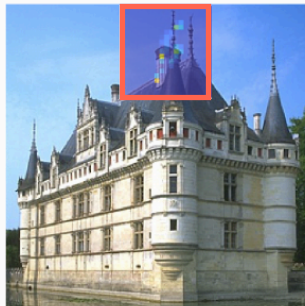
z

=



ϕ

*



$\bar{x}$

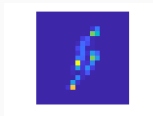
Basics about A: blur

$$z = A\bar{x} \quad \Leftrightarrow \quad z = \phi * \bar{x}$$



z

=



ϕ

*



$\bar{x}$

Basics about A: blur

$$z = A\bar{x} \quad \Leftrightarrow \quad z = \phi * \bar{x}$$



z

=



ϕ

*



$\bar{x}$

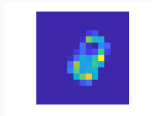
Basics about A: blur

$$z = A\bar{x} \quad \Leftrightarrow \quad z = \phi * \bar{x}$$



z

=



ϕ

*



$\bar{x}$

Basics about A: high-pass filtering

$$z = A\bar{x} \Leftrightarrow z = \phi * \bar{x}$$



z

$$= [-1, 1] *$$

ϕ



$\bar{x}$

Basics about A: high-pass filtering

$$z = A\bar{x} \Leftrightarrow z = \phi * \bar{x}$$



z

$$= [-1, 1]^T *$$

ϕ



$\bar{x}$

Hadamard conditions (1902)

The problem $z = A\bar{x}$ is said to be well-posed if it fulfills the **Hadamard conditions** :

1. **existence of a solution,**

i.e. the range $\text{range } A$ of A is equal to $\mathbb{R}^M$,

2. **uniqueness of the solution,**

i.e. the nullspace $\ker A$ of A is equal to $\{0\}$,

3. **stability of the solution $\hat{x}$ relatively to the observation,**

i.e. $(\forall (z, z') \in (\mathbb{R}^M)^2)$

$$\|z - z'\| \rightarrow 0 \quad \Rightarrow \quad \|\hat{x}(z) - \hat{x}(z')\| \rightarrow 0.$$

Hadamard conditions (1902)

The problem $z = A\bar{x}$ is said to be well-posed if it fulfills the **Hadamard conditions** :

1. **existence of a solution,**

i.e. every vector z in $\mathbb{R}^M$ is the image of a vector x in $\mathbb{R}^N$,

2. **uniqueness of the solution,**

i.e. if $\hat{x}(z)$ and $\hat{x}'(z)$ are two solutions, then they are necessarily equal since $\hat{x}(z) - \hat{x}'(z)$ belongs to $\ker A$,

3. **stability of the solution** $\hat{x}$ relatively to the observation,

i.e. ensure that a small perturbation of the observed image leads to a slight variation of the recovered image.

Inverse model $\hat{x} = d_{\Theta}(z)$ **when** $z = A\bar{x} + \varepsilon$

➔ [1922] **Maximum likelihood** (Fisher).

$$\begin{aligned}\hat{x} \in \underset{x}{\operatorname{Argmin}} \frac{1}{2} \|Ax - z\|_2^2 &= (A^*A)^{-1}A^*z = (A^*A)^{-1}A^*(A\bar{x} + \varepsilon) \\ &= \bar{x} + (A^*A)^{-1}A^*\varepsilon\end{aligned}$$

Inverse model $\hat{x} = d_{\Theta}(z)$ **when** $z = A\bar{x} + \varepsilon$

➔ [1922] **Maximum likelihood** (Fisher).

$$\begin{aligned}\hat{x} \in \underset{x}{\operatorname{Argmin}} \frac{1}{2} \|Ax - z\|_2^2 &= (A^*A)^{-1}A^*z = (A^*A)^{-1}A^*(A\bar{x} + \varepsilon) \\ &= \bar{x} + (A^*A)^{-1}A^*\varepsilon\end{aligned}$$

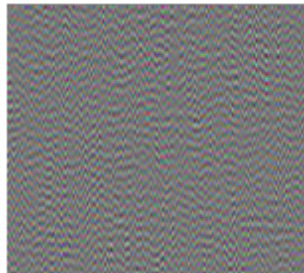
Noise amplification

Inverse model $\hat{x} = d_{\Theta}(z)$ **when** $z = A\bar{x} + \varepsilon$

→ [1922] **Maximum likelihood** (Fisher).

$$\begin{aligned}\hat{x} \in \underset{x}{\operatorname{Argmin}} \frac{1}{2} \|Ax - z\|_2^2 &= (A^*A)^{-1}A^*z = (A^*A)^{-1}A^*(A\bar{x} + \varepsilon) \\ &= \bar{x} + (A^*A)^{-1}A^*\varepsilon\end{aligned}$$

Noise amplification



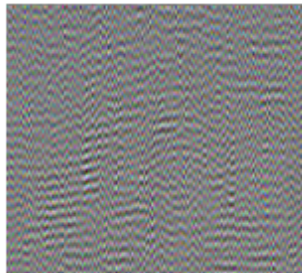
$$\varepsilon \sim \mathcal{N}(0, 0.05 \times \text{Id})$$

Inverse model $\hat{x} = d_{\Theta}(z)$ **when** $z = A\bar{x} + \varepsilon$

→ [1922] **Maximum likelihood** (Fisher).

$$\begin{aligned}\hat{x} \in \underset{x}{\operatorname{Argmin}} \frac{1}{2} \|Ax - z\|_2^2 &= (A^*A)^{-1}A^*z = (A^*A)^{-1}A^*(A\bar{x} + \varepsilon) \\ &= \bar{x} + (A^*A)^{-1}A^*\varepsilon\end{aligned}$$

Noise amplification



$$\varepsilon \sim \mathcal{N}(0, 0.01 \times \operatorname{Id})$$

Inverse model $\hat{x} = d_{\Theta}(z)$ **when** $z = A\bar{x} + \varepsilon$

→ [1922] **Maximum likelihood** (Fisher).

$$\begin{aligned}\hat{x} \in \underset{x}{\operatorname{Argmin}} \frac{1}{2} \|Ax - z\|_2^2 &= (A^*A)^{-1}A^*z = (A^*A)^{-1}A^*(A\bar{x} + \varepsilon) \\ &= \bar{x} + (A^*A)^{-1}A^*\varepsilon\end{aligned}$$

Noise amplification



$$\varepsilon \sim \mathcal{N}(0, 0.001 \times \operatorname{Id})$$

Inverse model $\hat{x} = d_{\Theta}(z)$ **when** $z = A\bar{x} + \varepsilon$

→ [1922] **Maximum likelihood** (Fisher).

$$\begin{aligned}\hat{x} \in \underset{x}{\operatorname{Argmin}} \frac{1}{2} \|Ax - z\|_2^2 &= (A^*A)^{-1}A^*z = (A^*A)^{-1}A^*(A\bar{x} + \varepsilon) \\ &= \bar{x} + (A^*A)^{-1}A^*\varepsilon\end{aligned}$$

Noise amplification



$$\varepsilon \sim \mathcal{N}(0, 0 \times \operatorname{Id})$$

Inverse model $\hat{x} = d_{\Theta}(z)$ **when** $z = A\bar{x} + \varepsilon$

➔ [1922] **Maximum likelihood** (Fisher).

$$\hat{x} \in \underset{x}{\operatorname{Argmin}} \frac{1}{2} \|Ax - z\|_2^2 = (A^*A)^{-1}A^*z$$

➔ [1963] **Regularisation** (Tikhonov, Huber)

$$\hat{x} \in \underset{x}{\operatorname{Argmin}} \frac{1}{2} \|Ax - z\|_2^2 + \theta \|Lx\|_2^2 \quad \text{avec } \theta > 0$$

➔ [2000] **Sparsity** (Donoho, Daubechies-Defrise-DeMol,...)

$$\hat{x} \in \underset{x}{\operatorname{Argmin}} \frac{1}{2} \|Ax - z\|_2^2 + \theta \|Lx\|_{\star}$$

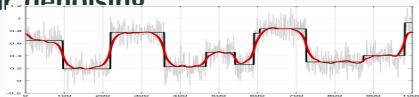
Regularized approaches

Piecewise constant denoising: $z = \bar{x} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \text{Id})$

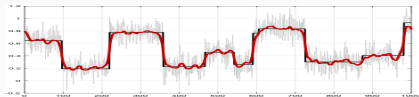
- Minimisation problem

$$\hat{x}(z; \hat{\theta}) = \arg \min_{x \in \mathbb{R}^N} \frac{1}{2} \|x - z\|_2^2 + \theta \|Lx\|_{\bullet} \quad \text{where} \quad \begin{cases} Lx = \psi * x \\ \theta > 0 \end{cases}$$

- Linear denoising



$$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix}; \quad \|\cdot\|_{\bullet} = \|\cdot\|_2^2; \quad \text{Large } \theta$$



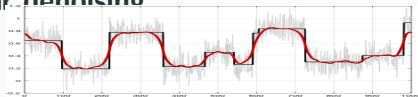
$$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix}; \quad \|\cdot\|_{\bullet} = \|\cdot\|_2^2; \quad \text{Small } \theta$$

Piecewise constant denoising: $z = \bar{x} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \text{Id})$

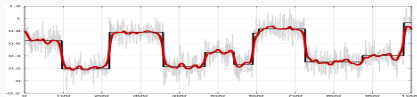
- Minimisation problem

$$\hat{x}(z; \hat{\theta}) = \arg \min_{x \in \mathbb{R}^N} \frac{1}{2} \|x - z\|_2^2 + \theta \|Lx\|_{\bullet} \quad \text{where} \quad \begin{cases} Lx = \psi * x \\ \theta > 0 \end{cases}$$

- Linear denoising



$$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix}; \quad \|\cdot\|_{\bullet} = \|\cdot\|_2^2; \quad \text{Large } \theta$$

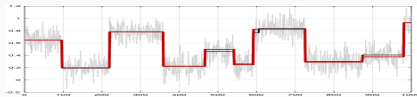


$$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix}; \quad \|\cdot\|_{\bullet} = \|\cdot\|_2^2; \quad \text{Small } \theta$$

- Nonlinear denoising



$$L = \begin{bmatrix} 1 & -1 \end{bmatrix} \quad \text{and} \quad \|\cdot\|_{\bullet} = \|\cdot\|_1$$



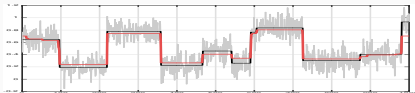
$$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix} \quad \text{and} \quad \|\cdot\|_{\bullet} = \|\cdot\|_0$$

Piecewise constant denoising: $z = \bar{x} + \varepsilon$ avec $\varepsilon = \mathcal{N}(0, \sigma^2 \text{Id})$

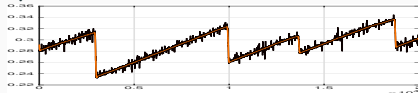
- Minimisation problem

$$\hat{x}(z; \hat{\theta}) = \arg \min_{x \in \mathbb{R}^N} \frac{1}{2} \|x - z\|_2^2 + \theta \|Lx\|_{\bullet} \quad \text{où} \quad \begin{cases} Lx = \psi * x \\ \theta > 0 \end{cases}$$

- Nonlinear denoising: piecewise constant/linear



$$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix} \quad \text{and} \quad \|\cdot\|_{\bullet} = \|\cdot\|_1$$



$$\psi = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \quad \text{and} \quad \|\cdot\|_{\bullet} = \|\cdot\|_1$$

Wavelets denoising: $z = \bar{x} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \text{Id})$

1 level wavelet decomposition



x



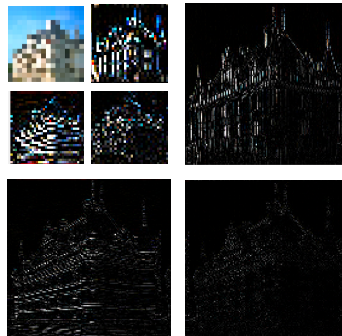
$\zeta = Lx$

Wavelets denoising: $z = \bar{x} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \text{Id})$

2 levels wavelet decomposition



x

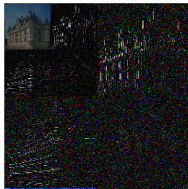


$\zeta = Lx$

Wavelets denoising: $z = \bar{x} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \text{Id})$



z



$\zeta = Lz$

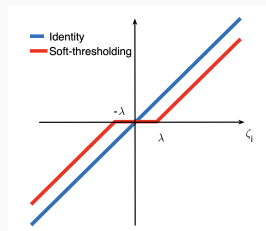


$\text{soft}_\theta(Lz)$



$\hat{x} = L^* \text{soft}_\theta(Lz)$

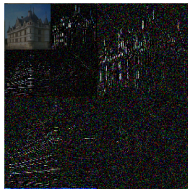
$$\begin{aligned}\text{soft}_\theta(\zeta) &= (\max\{|\zeta_{\underline{i}}| - \theta, 0\} \text{sgn}(\zeta_{\underline{i}}))_{\underline{i} \in \Omega} \\ &= \arg \min_{\nu} \frac{1}{2} \|\nu - \zeta\|_2^2 + \theta \|\nu\|_1\end{aligned}$$



Wavelets denoising: $z = \bar{x} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \text{Id})$



z



$\zeta = Lz$

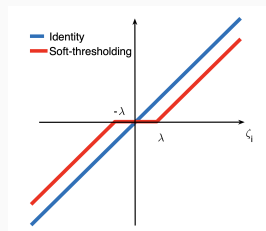


$\text{soft}_\theta(Lz)$



$\hat{x} = L^* \text{soft}_\theta(Lz)$

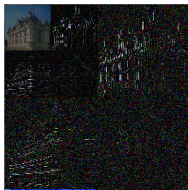
$$\begin{aligned}\text{soft}_\theta(\zeta) &= (\max\{|\zeta_{\underline{i}}| - \theta, 0\} \text{sgn}(\zeta_{\underline{i}}))_{\underline{i} \in \Omega} \\ &= \arg \min_{\nu} \frac{1}{2} \|\nu - \zeta\|_2^2 + \theta \|\nu\|_1\end{aligned}$$



Wavelets denoising: $z = \bar{x} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \text{Id})$



z



$\zeta = Lz$



$\text{soft}_\theta(Lz)$

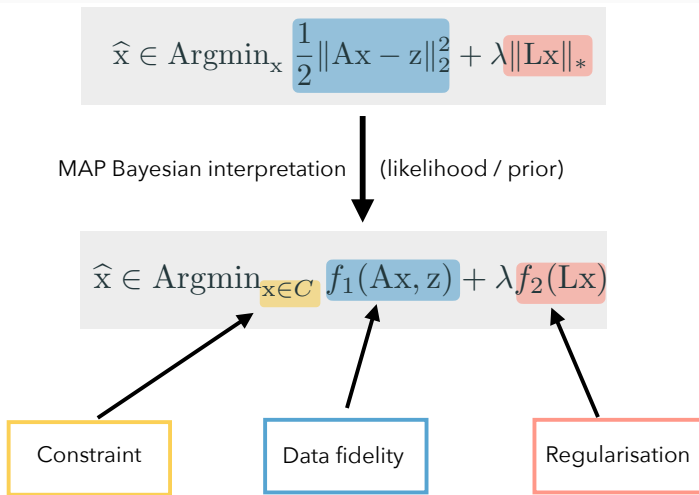


$\hat{x} = L^* \text{soft}_\theta(Lz)$

$$\begin{aligned}\text{soft}_\theta(\zeta) &= (\max\{|\zeta_{\underline{i}}| - \theta, 0\} \text{sgn}(\zeta_{\underline{i}}))_{\underline{i} \in \Omega} \\ &= \arg \min_{\nu} \frac{1}{2} \|\nu - \zeta\|_2^2 + \theta \|\nu\|_1\end{aligned}$$

$$\hat{x} = L^* \text{soft}_\theta(Lz) = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \theta \|L\mathbf{x}\|_1$$

Inverse model $\hat{x} = d_{\Theta}(z)$ with $z = A\bar{x} + \varepsilon$



Inverse model $\hat{x} = d_{\Theta}(z)$ **with** $z = A\bar{x} + \varepsilon$

$$\hat{x} \in \operatorname{Argmin}_x \frac{1}{2} \|Ax - z\|_2^2 + \lambda \|Lx\|_*$$

MAP Bayesian interpretation (likelihood / prior)

$$\hat{x} \in \operatorname{Argmin}_{x \in C} f_1(Ax, z) + \lambda f_2(Lx)$$

Minimisation algorithm

$$x^{[k+1]} = \mathbf{T}(x^{[k]})$$

Hyperparameter(s) estimation:

SURE, GCV, MCMC,...

Maximum A Posteriori (MAP)

Maximum A Posteriori (MAP)

Let $\mathbf{x}$ and $\mathbf{z}$ be random vector realizations X and Z .

$$\hat{\mathbf{x}} \in \underset{\mathbf{x} \in \mathbb{R}^N}{\text{Argmax}} \underbrace{\mu_{X|Z=\mathbf{z}}(\mathbf{x})}_{\text{Posterior distribution}}$$

Maximum A Posteriori (MAP)

Maximum A Posteriori (MAP)

Let $\mathbf{x}$ and $\mathbf{z}$ be random vector realizations X and Z .

$$\hat{\mathbf{x}} \in \underset{\mathbf{x} \in \mathbb{R}^N}{\text{Argmax}} \underbrace{\mu_{X|Z=\mathbf{z}}(\mathbf{x})}_{\text{Posterior distribution}}$$

Bayes rule:

$$\begin{aligned} \max_{\mathbf{x} \in \mathbb{R}^N} \mu_{X|Z=\mathbf{z}}(\mathbf{x}) &\Leftrightarrow \max_{\mathbf{x} \in \mathbb{R}^N} \mu_{Z|X=\mathbf{x}}(\mathbf{z}) \cdot \mu_X(\mathbf{x}) \\ &\Leftrightarrow \min_{\mathbf{x} \in \mathbb{R}^N} \left\{ -\log(\mu_{Z|X=\mathbf{x}}(\mathbf{z})) - \log(\mu_X(\mathbf{x})) \right\} \end{aligned}$$

Maximum A Posteriori (MAP)

Maximum A Posteriori (MAP)

Let $\mathbf{x}$ and $\mathbf{z}$ be random vector realizations X and Z .

$$\hat{\mathbf{x}} \in \underset{\mathbf{x} \in \mathbb{R}^N}{\text{Argmax}} \underbrace{\mu_{X|Z=\mathbf{z}}(\mathbf{x})}_{\text{Posterior distribution}}$$

Bayes rule:

$$\begin{aligned} \max_{\mathbf{x} \in \mathbb{R}^N} \mu_{X|Z=\mathbf{z}}(\mathbf{x}) &\Leftrightarrow \max_{\mathbf{x} \in \mathbb{R}^N} \mu_{Z|X=\mathbf{x}}(\mathbf{z}) \cdot \mu_X(\mathbf{x}) \\ &\Leftrightarrow \min_{\mathbf{x} \in \mathbb{R}^N} \left\{ \underbrace{-\log(\mu_{Z|X=\mathbf{x}}(\mathbf{z}))}_{\text{Data-term}} \underbrace{-\log(\mu_X(\mathbf{x}))}_{\text{A priori}} \right\} \\ &\Leftrightarrow \min_{\mathbf{x} \in \mathbb{R}^N} \mathbf{\textcolor{red}{f}}_1(\mathbf{x}) + \mathbf{\textcolor{blue}{f}}_2(\mathbf{x}) \end{aligned}$$

Data-fidelity term: Gaussian noise

$$(\forall \mathbf{x} \in \mathbb{R}^N) \quad f_1(\mathbf{x}) = -\log(\mu_{Z|X=\mathbf{x}}(z))$$

- Let $z = A\bar{x} + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \alpha \text{Id})$ with $\alpha > 0$.
- Gaussian likelihood:

$$\mu_{Z|X=\mathbf{x}}(z) = \prod_{n=1}^M \frac{1}{\sqrt{2\pi\alpha}} \exp\left(-\frac{((A\mathbf{x})_n - z_n)^2}{2\alpha}\right)$$

- Data-term:

$$f_1(\mathbf{x}) = \sum_{n=1}^M \frac{1}{2\alpha} ((A\mathbf{x})_n - z_n)^2$$

Data-fidelity term: Poisson noise

$$(\forall \mathbf{x} \in \mathbb{R}^N) \quad f_1(\mathbf{x}) = -\log(\mu_{Z|X=\mathbf{x}}(z))$$

- Let $z = \mathcal{D}_\alpha(A\bar{\mathbf{x}})$ where $\mathcal{D}_\alpha$ Poisson noise with parameter α .

- Poisson likelihood:

$$\mu_{Z|X=\mathbf{x}}(z) = \prod_{n=1}^M \frac{\exp(-\alpha(A\mathbf{x})_n)}{z_n!} (\alpha(A\mathbf{x})_n)^{z_n}$$

- Data-term: $f_1(\mathbf{x}) = \sum_{n=1}^M \Psi_i((A\mathbf{x})_n)$

$$(\forall v \in \mathbb{R}) \quad \Psi_i(v) = \begin{cases} \alpha v - z_n \log(\alpha v) & \text{if } z_n > 0 \text{ and } v > 0, \\ \alpha v & \text{si } z_n = 0 \text{ and } v \geq 0, \\ +\infty & \text{otherwise.} \end{cases}$$

**Subdifferential,
Proximity operator,
Proximal algorithms**

From smooth to non-smooth optimization to solve $\hat{\mathbf{x}} \in \underset{\mathbf{x}}{\text{Argmin}} f(\mathbf{x})$

Gradient descent

$$\mathbf{x}^{[k+1]} = \mathbf{x}^{[k]} - \tau \nabla f(\mathbf{x}^{[k]})$$

(Forward) subgradient descent

$$\mathbf{x}^{[k+1]} = \mathbf{x}^{[k]} - \mathbf{u}^{[k]} \quad \text{where} \quad \mathbf{u}^{[k]} \in \tau \partial f(\mathbf{x}^{[k]})$$

(Backward) subgradient descent

$$\mathbf{x}^{[k+1]} = \mathbf{x}^{[k]} - \mathbf{u}^{[k]} \quad \text{where} \quad \mathbf{u}^{[k]} \in \tau \partial f(\mathbf{x}^{[k+1]})$$

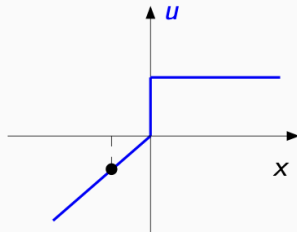
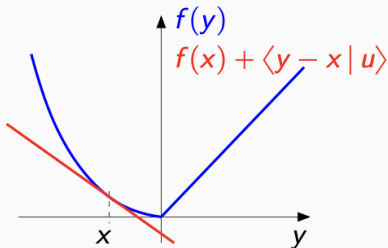
Subdifferential

Let $f : \mathcal{H} \rightarrow \mathbb{R}$ be a proper function.

The **Moreau subdifferential of f** , denoted ∂f , is such that:

$$\partial f : \mathcal{H} \rightarrow 2^{\mathcal{H}}$$

$$x \rightarrow \{u \in \mathcal{H} \mid (\forall y \in \mathcal{H}) \langle y - x | u \rangle + f(x) \leq f(y)\}$$



Subdifferential

Let $f : \mathcal{H} \rightarrow \mathbb{R}$ be a proper function.

The **Moreau subdifferential of f** , denoted ∂f , is such that:

$$\begin{aligned}\partial f : \mathcal{H} &\rightarrow 2^{\mathcal{H}} \\ \mathbf{x} &\rightarrow \{\mathbf{u} \in \mathcal{H} \mid (\forall \mathbf{y} \in \mathcal{H}) \langle \mathbf{y} - \mathbf{x} | \mathbf{u} \rangle + f(\mathbf{x}) \leq f(\mathbf{y})\}\end{aligned}$$

Fermat rule: $0 \in \partial f(\hat{\mathbf{x}}) \Leftrightarrow \hat{\mathbf{x}} \in \underset{\mathbf{x}}{\operatorname{Argmin}} f(\mathbf{x})$

Forward subgradient descent

$$\mathbf{x}^{[k+1]} = \mathbf{x}^{[k]} - \mathbf{u}^{[k]} \quad \text{where } \mathbf{u}^{[k]} \in \tau \partial f(\mathbf{x}^{[k]})$$

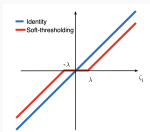
Proximity operator

Definition [Moreau,1965] Let $f: \mathcal{H} \rightarrow]-\infty, +\infty]$ be a proper, l.s.c, and convex function. The proximity operator of f at point $x \in \mathcal{H}$ is the **unique point** denoted $\text{prox}_f x$ such that

$$(\forall x \in \mathcal{H}) \quad \text{prox}_f x = \arg \min_{v \in \mathcal{H}} \frac{1}{2} \|x - v\|^2 + f(v)$$

➔ Numerous closed forms are available

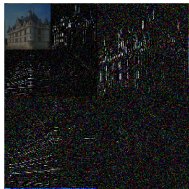
- $\text{prox}_{\iota_C} = P_C$: **Projection** onto a convex set.
- $\text{prox}_{\theta \|\cdot\|_1}$: **soft-thresholding** with threshold $\theta > 0$.
- Full list available: **PROX Repository**



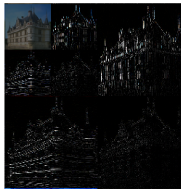
Wavelets denoising: $z = \bar{x} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \text{Id})$



z



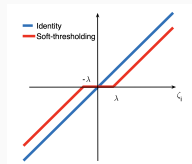
$\zeta = \Psi z$



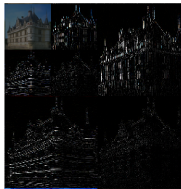
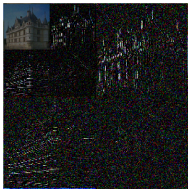
$\text{soft}_\theta(\Psi z)$



$\hat{x} = \Psi^* \text{soft}_\theta(\Psi z)$



Wavelets denoising: $z = \bar{x} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \text{Id})$



z

$$\zeta = \Psi z$$

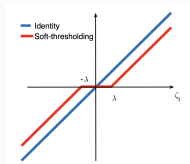
$$\text{soft}_{\theta}(\Psi z)$$

$$\hat{x} = \Psi^* \text{soft}_{\theta}(\Psi z)$$

$$\text{soft}_{\theta}(\zeta) = \left(\max\{|\zeta_i| - \theta, 0\} \text{sgn}(\zeta_i) \right)_{i \in \Omega}$$

$$= \arg \min_{\nu} \frac{1}{2} \|\nu - \zeta\|_2^2 + \theta \|\nu\|_1$$

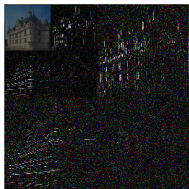
$$= \text{prox}_{\theta \|\cdot\|_1}(\zeta) \rightarrow \text{proximity operator}$$



Wavelets denoising: $z = \bar{x} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \text{Id})$



z



$\zeta = \Psi z$



$\text{soft}_\theta(\Psi z)$



$\hat{x} = \Psi^* \text{soft}_\theta(\Psi z)$

$$\hat{x} = \Psi^* \text{soft}_\theta(\Psi z)$$

$$= \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \theta \|\Psi \mathbf{x}\|_1 = \text{prox}_{\theta \|\Psi \cdot\|_1}(\mathbf{z})$$

Proximity operator

Let $\mathcal{H}$ be a Hilbert space and $f \in \Gamma_0(\mathcal{H})$.

$$(\forall x \in \mathcal{H}) \quad p = \text{prox}_f(x) \quad \Leftrightarrow \quad x - p \in \partial f(p).$$

Proximity operator

Let $\mathcal{H}$ be a Hilbert space and $f \in \Gamma_0(\mathcal{H})$.

$$(\forall \mathbf{x} \in \mathcal{H}) \quad \mathbf{p} = \text{prox}_f(\mathbf{x}) \quad \Leftrightarrow \quad \mathbf{x} - \mathbf{p} \in \partial f(\mathbf{p}).$$

• Proof:

$$\begin{aligned} \mathbf{p} = \arg \min_{\mathbf{y} \in \mathcal{H}} f(\mathbf{y}) + \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|^2 &\quad \Leftrightarrow \quad 0 \in \partial \left(f + \frac{1}{2} \|\cdot - \mathbf{x}\|^2 \right) (\mathbf{p}) \\ &\quad \Leftrightarrow \quad 0 \in \partial f(\mathbf{p}) + \mathbf{p} - \mathbf{x} \end{aligned}$$

Proximity operator

Let $\mathcal{H}$ be a Hilbert space and $f \in \Gamma_0(\mathcal{H})$.

$$(\forall x \in \mathcal{H}) \quad p = \text{prox}_f(x) \quad \Leftrightarrow \quad x - p \in \partial f(p).$$

• Proof:

$$\begin{aligned} p = \arg \min_{y \in \mathcal{H}} f(y) + \frac{1}{2} \|y - x\|^2 &\quad \Leftrightarrow \quad 0 \in \partial \left(f + \frac{1}{2} \|\cdot - x\|^2 \right)(p) \\ &\quad \Leftrightarrow \quad 0 \in \partial f(p) + p - x \end{aligned}$$

Proximity operator

Let $\mathcal{H}$ be a Hilbert space and $f \in \Gamma_0(\mathcal{H})$.

$$(\forall x \in \mathcal{H}) \quad p = \text{prox}_f(x) \quad \Leftrightarrow \quad x - p \in \partial f(p).$$

Backward subgradient descent = **Proximal point algorithm**

$$\begin{aligned} x^{[k+1]} &= x^{[k]} - u^{[k]} \quad \text{where } u^{[k]} \in \tau \partial f(x^{[k+1]}) \\ &= \text{prox}_{\tau f}(x^{[k]}) \end{aligned}$$

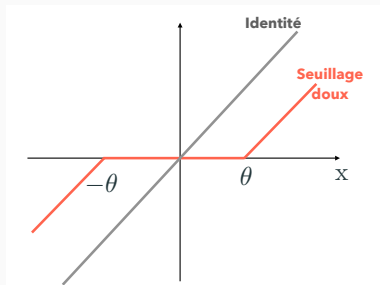
Proximal algorithm to solve $\hat{\mathbf{x}} \in \underset{\mathbf{x}}{\operatorname{Argmin}} f(\mathbf{x})$

Key tool – *Proximity operator*

Backward subgradient descent:

$$\begin{aligned} \mathbf{x}^{[k+1]} &= \mathbf{x}^{[k]} - \mathbf{u}^{[k]} \quad \text{où } \mathbf{u}^{[k]} \in \tau \partial f(\mathbf{x}^{[k+1]}) \\ &= \operatorname{prox}_{\tau f}(\mathbf{x}^{[k]}) \end{aligned}$$

➔ Proximity operator of the ℓ_1 norm = soft-thresholding



Proximal algorithm to solve $\hat{\mathbf{x}} \in \underset{\mathbf{x}}{\text{Argmin}} f(\mathbf{x})$

Key tool – Proximity operator

Backward subgradient descent:

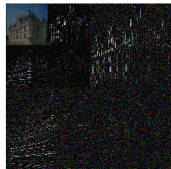
$$\begin{aligned} \mathbf{x}^{[k+1]} &= \mathbf{x}^{[k]} - \mathbf{u}^{[k]} \quad \text{où } \mathbf{u}^{[k]} \in \tau \partial f(\mathbf{x}^{[k+1]}) \\ &= \text{prox}_{\tau f}(\mathbf{x}^{[k]}) \end{aligned}$$

➔ Proximity operator of the ℓ_1 norm = soft-thresholding

➔ Denoising (L orthon. basis): $\hat{\mathbf{x}} = \text{prox}_{\tau\theta\|\cdot\|_1}(\mathbf{z}) = \mathbf{L}^* \text{soft}_{\gamma\theta}(\mathbf{L}\mathbf{z})$



$\mathbf{z}$



$\mathbf{u} = \mathbf{L}\mathbf{z}$



$\text{soft}_{\tau\theta}(\mathbf{L}\mathbf{z})$



$\hat{\mathbf{x}} = \mathbf{L}^* \text{soft}_{\theta}(\mathbf{L}\mathbf{z})$

Proximal algorithm to solve $\hat{\mathbf{x}} \in \underset{\mathbf{x}}{\text{Argmin}} f(\mathbf{x})$

Key tool – *Proximity operator*

Backward subgradient descent:

$$\begin{aligned} \mathbf{x}^{[k+1]} &= \mathbf{x}^{[k]} - \mathbf{u}^{[k]} \quad \text{où } \mathbf{u}^{[k]} \in \tau \partial f(\mathbf{x}^{[k+1]}) \\ &= \text{prox}_{\tau f}(\mathbf{x}^{[k]}) \end{aligned}$$

- ➔ Proximity operator of the ℓ_1 norm = soft-thresholding
- ➔ Denoising (L orthon. basis): $\hat{\mathbf{x}} = \text{prox}_{\tau \theta \|L \cdot\|_1}(\mathbf{z}) = L^* \text{soft}_{\gamma \theta}(L\mathbf{z})$
- ➔ Numerous closed forms: <http://proximity-operator.net>.
- ➔ ... but also **more complex operations**: $\text{prox}_{f_1+f_2}$.

Proximal algorithm to solve $\hat{\mathbf{x}} \in \underset{\mathbf{x}}{\operatorname{Argmin}} f(\mathbf{x})$

Key tool – *Proximity operator*

Backward subgradient descent:

$$\begin{aligned} \mathbf{x}^{[k+1]} &= \mathbf{x}^{[k]} - \mathbf{u}^{[k]} \quad \text{où } \mathbf{u}^{[k]} \in \tau \partial f(\mathbf{x}^{[k+1]}) \\ &= \operatorname{prox}_{\tau f}(\mathbf{x}^{[k]}) \end{aligned}$$

➔ Proximity operator of the ℓ_1 norm = soft-thresholding

➔ Denoising (L orthon. basis): $\hat{\mathbf{x}} = \operatorname{prox}_{\tau \theta \|L \cdot\|_1}(\mathbf{z}) = L^* \operatorname{soft}_{\gamma \theta}(L\mathbf{z})$

➔ Numerous closed forms: <http://proximity-operator.net>.

➔ ... but also **more complex operations**: $\operatorname{prox}_{f_1+f_2}$.

Forward-backward algorithm – $\hat{\mathbf{x}} \in \underset{\mathbf{x}}{\operatorname{Argmin}} f_1(\mathbf{x}) + f_2(\mathbf{x})$

$$\mathbf{x}^{[k+1]} = \operatorname{prox}_{\tau f_2}(\mathbf{x}^{[k]} - \tau \nabla f_1(\mathbf{x}^{[k]}))$$

→ Minimisation problem :

$$\hat{x} \in \underset{x}{\operatorname{Argmin}} f_1(x) + f_2(x)$$

with f_1 and f_2 either diff. with Lipschitz gradient or proximable.

→ Design of a recursive sequence of the form:

$$(\forall k \in \mathbb{N}) \quad x^{[k+1]} = \mathbf{T}x^{[k]},$$

Gradient descent	$\mathbf{T} = \operatorname{Id} - \tau(\nabla f_1 + \nabla f_2)$
Proximal point	$\mathbf{T} = \operatorname{prox}_{\tau(f_1+f_2)}$
Forward-Backward	$\mathbf{T} = \operatorname{prox}_{\tau f_2}(\operatorname{Id} - \tau \nabla f_1)$
Peaceman-Rachford	$\mathbf{T} = (2\operatorname{prox}_{\tau f_2} - \operatorname{Id}) \circ (2\operatorname{prox}_{\tau f_1} - \operatorname{Id})$
Douglas-Rachford	$\mathbf{T} = \operatorname{prox}_{\tau f_2}(2\operatorname{prox}_{\tau f_1} - \operatorname{Id}) + \operatorname{Id} - \operatorname{prox}_{\tau f_1}$

General objective function

$$\text{Find} \quad \hat{\mathbf{x}} \in \underset{\mathbf{x} \in \mathcal{H}}{\text{Argmin}} \sum_{j=1}^J f_j(\mathbf{H}_j \mathbf{x})$$

where $\mathbf{H}_j$ denotes a linear operator from $\mathcal{H}$ to $\mathcal{G}_j$ and $(f_j)_{1 \leq j \leq J}$ belong to the class of convex, l.s.c., and proper from $\mathcal{G}_j$ to $] - \infty, +\infty]$.

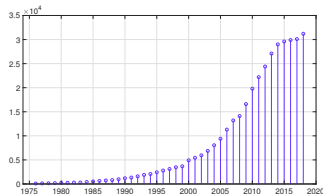
Proximal algorithms

General objective function

$$\text{Find } \hat{x} \in \underset{x \in \mathcal{H}}{\text{Argmin}} \sum_{j=1}^J f_j(H_j x)$$

where H_j denotes a linear operator from $\mathcal{H}$ to $\mathcal{G}_j$ and $(f_j)_{1 \leq j \leq J}$ belong to the class of convex, l.s.c., and proper from $\mathcal{G}_j$ to $] - \infty, +\infty]$.

- Numerous proximal algorithms
 - Forward-Backward
 - Douglas-Rachford
 - ADMM
 - Primal-dual ...

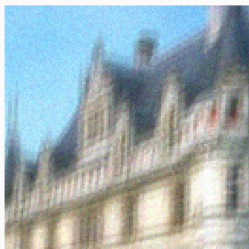


Number of articles per year on Google scholar containing "proximal algorithms" since 1997.

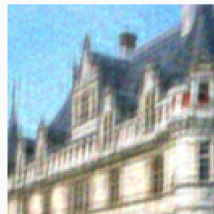
Evolution of image restoration results: blur + Gaussian noise



Original



Degraded (13.4 dB)



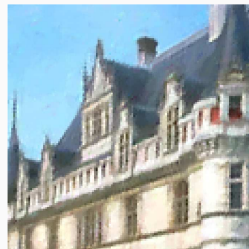
Tikhonov (16.4 dB)



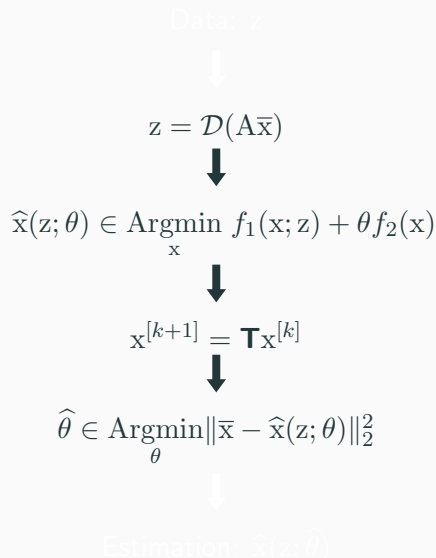
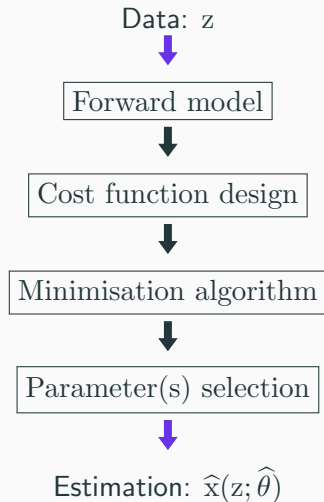
DTT (16.6 dB)



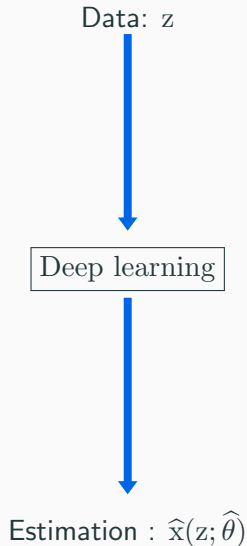
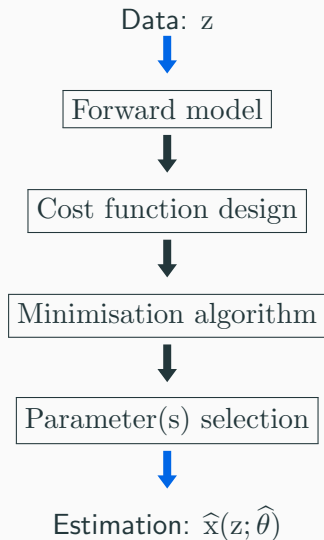
TV (19.0 dB)



NLTV (19.4 dB)



Towards deep learning



Inversion model $\hat{x} = d_{\Theta}(z)$

→ [1922] **Maximum likelihood** (Fisher).

$$\hat{x} \in \underset{x}{\operatorname{Argmin}} \frac{1}{2} \|Ax - z\|_2^2 = (A^*A)^{-1}A^*z$$

→ [1963] **Regularisation** (Tikhonov, Huber)

$$\hat{x} \in \underset{x}{\operatorname{Argmin}} \frac{1}{2} \|Ax - z\|_2^2 + \theta \|Lx\|_2^2 \quad \text{avec } \theta > 0$$

→ [2000] **Sparsity** (Donoho, Daubechies-Defrise-DeMol,...)

$$\hat{x} \in \underset{x}{\operatorname{Argmin}} \frac{1}{2} \|Ax - z\|_2^2 + \theta \|Lx\|_{\star}$$

→ [2010] **“End to end” neural networks** $\hat{x} = \operatorname{NN}_{\Theta}(z)$

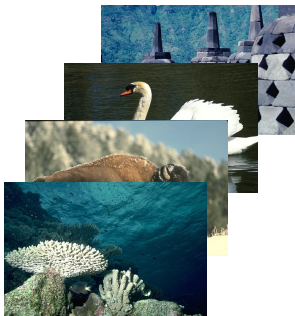
→ [2020] **Model-based neural network: PnP, unrolled, ...**

$$0 \in A^*(A\hat{x} - z) + \mathbf{B}(\hat{x})$$

Inversion model $\hat{x} = d_{\Theta}(z)$

Deep learning – General framework

- Dataset : $\mathcal{S} = \{(\bar{x}_{\ell}, z_{\ell}) \in \mathbb{R}^N \times \mathbb{R}^M \mid \ell \in \mathbb{I} \cup \mathbb{J}\}$



$\bar{x}_{\ell}$



z_{ℓ}

Inversion model $\hat{x} = d_{\Theta}(z)$

Deep learning – General framework

- Dataset : $\mathcal{S} = \{(\bar{x}_{\ell}, z_{\ell}) \in \mathbb{R}^N \times \mathbb{R}^M \mid \ell \in \mathbb{I} \cup \mathbb{J}\}$
 - *training set* $(\bar{x}_{\ell}, z_{\ell})_{\ell \in \mathbb{I}}$ with size $\mathbb{I}$
 - *testing set* $(\bar{x}_{\ell}, z_{\ell})_{\ell \in \mathbb{J}}$ with size $\mathbb{J}$

Inversion model $\hat{\mathbf{x}} = \mathbf{d}_{\Theta}(\mathbf{z})$

Deep learning – General framework

- Dataset : $\mathcal{S} = \{(\bar{\mathbf{x}}_{\ell}, \mathbf{z}_{\ell}) \in \mathbb{R}^N \times \mathbb{R}^M \mid \ell \in \mathbb{I} \cup \mathbb{J}\}$
 - *training set* $(\bar{\mathbf{x}}_{\ell}, \mathbf{z}_{\ell})_{\ell \in \mathbb{I}}$ with size $\mathbb{I}$
 - *testing set* $(\bar{\mathbf{x}}_{\ell}, \mathbf{z}_{\ell})_{\ell \in \mathbb{J}}$ with size $\mathbb{J}$
- Prediction function: $\mathbf{d}_{\Theta}(\mathbf{z}_{\ell}) = \eta^{[K]}(W^{[K]} \dots \eta^{[1]}(W^{[1]} \mathbf{z}_{\ell} + \mathbf{b}^{[1]}) \dots + \mathbf{b}^{[K]})$
 - Linear operators: $W^{[1]}, W^{[2]}, \dots, W^{[K]}$
 - Activation functions: $\eta^{[1]}, \eta^{[2]}, \dots, \eta^{[K]}$
 - Bias: $\mathbf{b}^{[1]}, \mathbf{b}^{[2]}, \dots, \mathbf{b}^{[K]}$

$$\Rightarrow \Theta = \{W^{[1]}, \dots, W^{[K]}, \mathbf{b}^{[1]}, \dots, \mathbf{b}^{[K]}\}$$

Inversion model $\hat{\mathbf{x}} = \mathbf{d}_{\Theta}(\mathbf{z})$

Deep learning – General framework

- Dataset : $\mathcal{S} = \{(\bar{\mathbf{x}}_{\ell}, \mathbf{z}_{\ell}) \in \mathbb{R}^N \times \mathbb{R}^M \mid \ell \in \mathbb{I} \cup \mathbb{J}\}$
 - *training set* $(\bar{\mathbf{x}}_{\ell}, \mathbf{z}_{\ell})_{\ell \in \mathbb{I}}$ with size $\mathbb{I}$
 - *testing set* $(\bar{\mathbf{x}}_{\ell}, \mathbf{z}_{\ell})_{\ell \in \mathbb{J}}$ with size $\mathbb{J}$
- Prediction function: $\mathbf{d}_{\Theta}(\mathbf{z}_{\ell}) = \eta^{[K]}(W^{[K]} \dots \eta^{[1]}(W^{[1]}\mathbf{z}_{\ell} + \mathbf{b}^{[1]}) \dots + \mathbf{b}^{[K]})$
 $\Rightarrow \Theta = \{W^{[1]}, \dots, W^{[K]}, \mathbf{b}^{[1]}, \dots, \mathbf{b}^{[K]}\}$
- *Learn parameters*: $\hat{\Theta} \in \text{Argmin}_{\Theta} \frac{1}{\mathbb{I}} \sum_{\ell \in \mathbb{I}} \mathcal{L}(\bar{\mathbf{x}}_{\ell}, \mathbf{d}_{\Theta}(\mathbf{z}_{\ell}))$

Inversion model $\hat{\mathbf{x}} = \mathbf{d}_{\Theta}(\mathbf{z})$

Deep learning – General framework

- Dataset : $\mathcal{S} = \{(\bar{\mathbf{x}}_{\ell}, \mathbf{z}_{\ell}) \in \mathbb{R}^N \times \mathbb{R}^M \mid \ell \in \mathbb{I} \cup \mathbb{J}\}$
 - *training set* $(\bar{\mathbf{x}}_{\ell}, \mathbf{z}_{\ell})_{\ell \in \mathbb{I}}$ with size $\mathbb{I}$
 - *testing set* $(\bar{\mathbf{x}}_{\ell}, \mathbf{z}_{\ell})_{\ell \in \mathbb{J}}$ with size $\mathbb{J}$
- Prediction function: $\mathbf{d}_{\Theta}(\mathbf{z}_{\ell}) = \eta^{[K]}(W^{[K]} \dots \eta^{[1]}(W^{[1]}\mathbf{z}_{\ell} + \mathbf{b}^{[1]}) \dots + \mathbf{b}^{[K]})$
 $\Rightarrow \Theta = \{W^{[1]}, \dots, W^{[K]}, \mathbf{b}^{[1]}, \dots, \mathbf{b}^{[K]}\}$
- *Learn parameters*: $\hat{\Theta} \in \text{Argmin}_{\Theta} \frac{1}{\mathbb{I}} \sum_{\ell \in \mathbb{I}} \mathcal{L}(\bar{\mathbf{x}}_{\ell}, \mathbf{d}_{\Theta}(\mathbf{z}_{\ell}))$
- *Evaluate*: A properly trained network must satisfy

$$(\forall \ell \in \mathbb{J}) \quad \bar{\mathbf{x}}_{\ell} \approx \mathbf{d}_{\hat{\Theta}}(\mathbf{z}_{\ell})$$

Inversion model $\hat{x} = d_{\Theta}(z)$: Plug-and-Play (PnP)

Deep learning – General framework

- Dataset : $\mathcal{S} = \{(\bar{x}_i, z_i) \in \mathbb{R}^N \times \mathbb{R}^M \mid i \in \{1, \dots, \mathbb{I}\}\}$
- Prediction function : $\text{NN}_{\Theta}(z_i) = \eta^{[K]}(W^{[K]} \dots \eta^{[1]}(W^{[1]}z_i + b^{[1]}) \dots + b^{[K]})$

Variational versus plug-and-play approach

- Analysis variational approach : $\min_x \frac{1}{2} \|Ax - z\|_2^2 + \theta \|Lx\|_{\star}$
- Forward-backward algorithm:

$$x^{[k+1]} = \text{prox}_{\tau\theta\|L\cdot\|_{\star}} (x^{[k]} - \tau A^*(Ax^{[k]} - z))$$

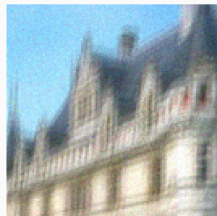
- **Plug-and-Play algorithm:**

$$x^{[k+1]} = \text{NN}_{\Theta} (x^{[k]} - \tau A^*(Ax^{[k]} - z))$$

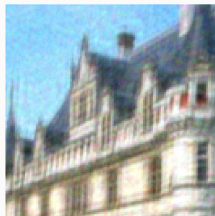
Performance summary



Original



Degraded
 $\text{SNR} = 13.4 \text{ dB}$



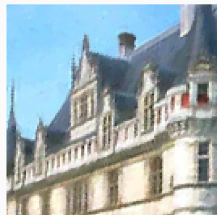
Tikhonov
 $\text{SNR} = 16.4 \text{ dB}$



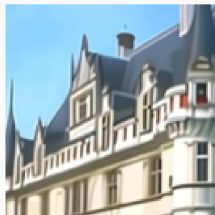
DTT
 $\text{SNR} = 16.6 \text{ dB}$



TV



NLTV



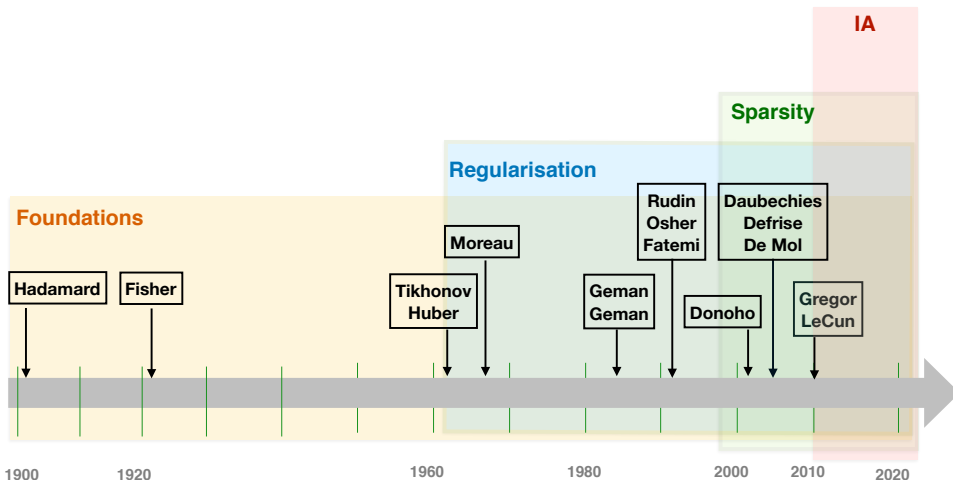
PnP-DRUnet



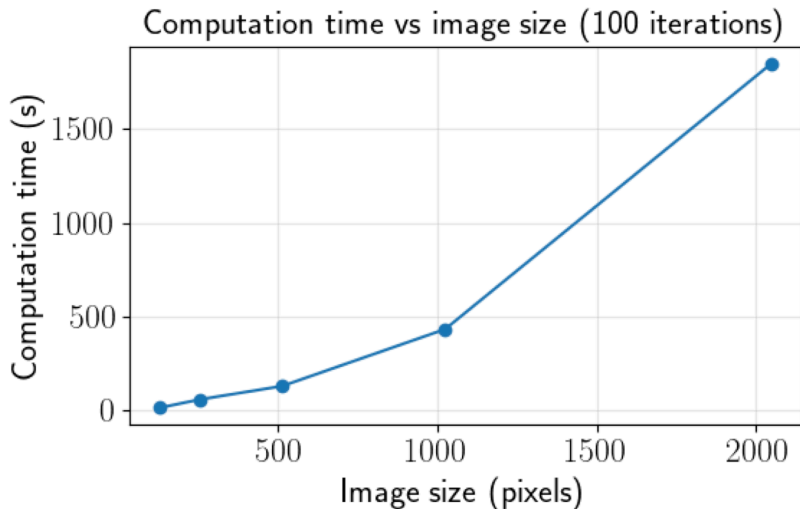
PnP-ScCP

Summary of the history of inverse problems

(Focus on reconstruction methods)



Main bottleneck: scale with the dimension of the image



How to accelerate the convergence?

Several possibilities:

- Use higher-order/preconditioned methods: $x^{[k+1]} = x^{[k]} + \alpha^{[k]} P^{-1} \nabla f(x^{[k]})$
 - Less iterations but higher cost per iteration
 - Difficult to build a good preconditioner
- Use inertial methods
 - Depending on the problem may reduce the number of iterations
- Use block-coordinate descent methods
 - How to build blocks for imaging problems?
- Exploit multiresolution!
 - Can be coupled with all previous methods!

Multiresolution to accelerate optimization schemes

Origin: multigrid methods for the numerical solution of linear PDEs

Given $f : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}$, $g : \partial\Omega \rightarrow \mathbb{R}$
and a linear differential operator D , find
 $u : \mathbb{R}^d \rightarrow \mathbb{R}$ s.t.

$$D(u(x)) = f(x) \text{ in } \Omega$$

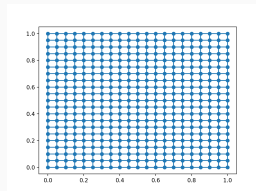
$$u(x) = g(x) \text{ in } \partial\Omega$$



Discretization



Linear system $Au = f$



$d=2$, $\Omega = [0, 1] \times [0, 1]$

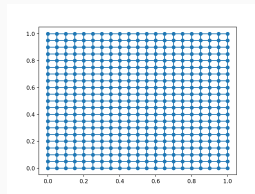
The size of the grid impacts:

- the **size** of the discrete problem
- the **accuracy** of the solution approximation

Multigrid methods

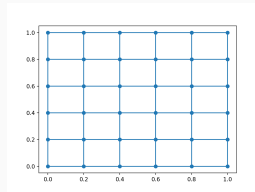
Aim: accelerate fixed point methods by exploiting multiple grids

- Fine grid: $A^h u^h = f^h$

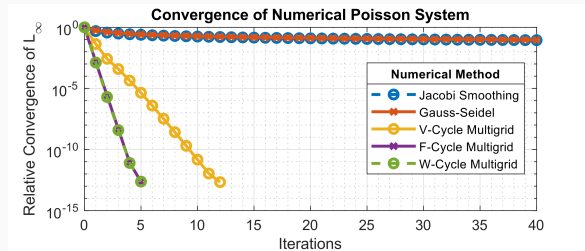
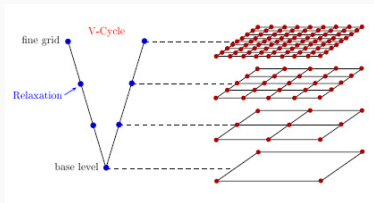


$R \downarrow$ $P \uparrow$

- Coarse grid: $A^H u^H = f^H$



General multigrid methods



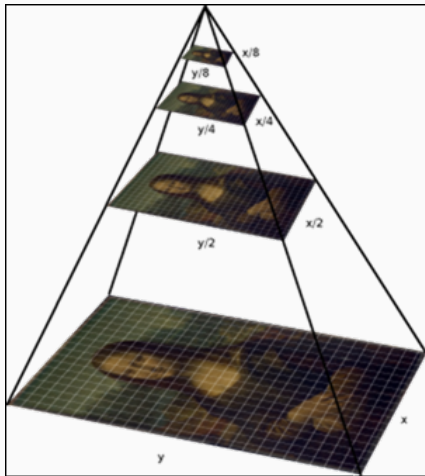
[W. L. Briggs, V. E. Henson, and S. F. McCormick **A Multigrid Tutorial**, 2000]

From linear to nonlinear: multilevel methods

Multigrid methods work for linear systems: $Ax = b$

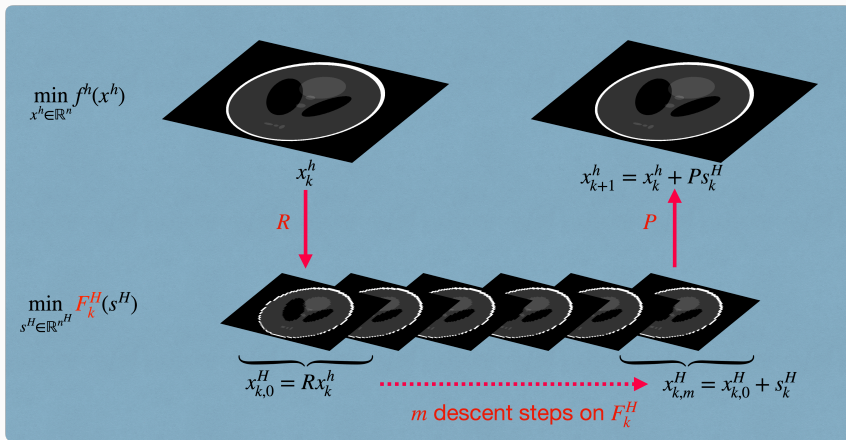
Multilevel schemes extend this paradigm to **optimization problem** with a hierarchical structure

$$\min_x f(x)$$



Multilevel methods: key idea

Alternate accurate "fine" steps and cheap "coarse" steps on reduced dimensional image



Challenge: correctly define R, P, F_k^H

Outline

1. 09/09 - Basics on smooth optimization
2. 14/09 - Acceleration methods: momentum, second order
3. 16/09 - Globalization strategies
4. 21/09 - Acceleration methods: multigrid and multilevel (smooth)
5. 23/09 - Acceleration methods: multilevel (convergence)
6. 28/09 - no course
7. 30/09 - Convex non-smooth optim: basics
8. 08/10 - Convex non-smooth optim: algorithms
9. 07/10 - Wavelets and image restoration
10. 12/10 - Non smooth multilevel
11. 14/10 - Non smooth multilevel and link with BCD methods
12. 19/10 - Second order proximal methods and preconditioning
13. 21/10 - Neural networks in imaging
14. 02/11 - Plug and Play and unfolded
15. 04/11 - Primal-dual algorithms

- CC: questions proposed at the end of each course to work at home. Last 20 minutes of each course dedicated to interview 2 students chosen randomly on the questions worked at home
- Exam: oral presentation of articles