

# Algorithme multiniveau hybride pour la restauration d’images

Nils LAURENT<sup>1</sup> Elisa RICCIETTI<sup>2</sup> Julian TACHELLA<sup>1</sup> Nelly PUSTELNIK<sup>1</sup>

<sup>1</sup>Ens de Lyon, CNRS, Laboratoire de Physique, F-69342, Lyon, France.

<sup>2</sup>ENS de Lyon, CNRS, Inria, Université Claude Bernard Lyon 1, LIP, UMR 5668, 69342, Lyon cedex 07, France

**Résumé** – Cet article présente un algorithme Plug-and-Play avec une structure multiniveau ayant pour objectif d’accélérer la reconstruction d’une image dégradée. En effet, la stratégie multiniveau s’est avérée efficace pour accélérer les algorithmes dans des contextes variés et de récents travaux ont adapté ce cadre dans le contexte Plug-and-Play. L’implémentation qui a été proposée repose cependant à tous les niveaux sur un réseau qui est coûteux à évaluer. C’est pourquoi nous proposons d’intégrer dans les niveaux grossiers de l’algorithme des itérations variationnelles, qui simplifient fortement le calcul et qui accélèrent davantage la reconstruction.

**Abstract** – This article presents a Plug-and-Play algorithm with a multilevel structure, aimed at accelerating image reconstruction. Indeed, the multilevel strategy has proven its efficiency in accelerating algorithms in multiple contexts and recent works have extended them to the Plug-and-Play context. The proposed implementation relies however on a neural network that is expensive to evaluate. We thus propose to integrate in the coarse levels of the algorithm variational iterations to strongly simplify the computations and thus further accelerate the reconstruction.

## 1 Introduction

La restauration d’images est essentielle dans de nombreux domaines scientifiques, comme le domaine médical ou la physique. Dans ces travaux, nous nous intéressons à la reconstruction par la résolution des problèmes inverses de la forme

$$z = A\bar{x} + \varepsilon \quad (1)$$

où  $\bar{x} \in \mathbb{R}^N$  est l’image originale dégradée par l’opérateur  $A \in \mathbb{R}^{M \times N}$ , et bruitée aléatoirement par  $\varepsilon$ . Le problème (1) est généralement mal conditionné et les algorithmes de résolutions itératifs sont généralement fondés sur deux opérations successives : une première qui guide la solution pour être en adéquation avec le modèle d’observation (cf. (1)) et une deuxième qui permet à la solution de satisfaire un modèle *a priori*. Si pendant de nombreuses années, l’*a priori* était défini par l’utilisateur à partir d’hypothèses sous-jacentes sur la classe d’image à reconstruire, les méthodes d’apprentissages ont permis de rebattre les cartes de cette approche en imposant un *a priori* implicite appris (via son action en tant que débruiteur), permettant d’importants gains sur la qualité de reconstruction [14]. Dans la littérature, deux approches d’apprentissage ressortent : l’approche par *algorithme déroulé*, qui consiste à réaliser un entraînement propre au type de dégradation et vise à reconstruire l’image en un faible nombre d’itérations [7], et l’approche *Plug-and-Play* (PnP), qui permet d’intégrer dans l’algorithme un débruiteur obtenu par apprentissage profond, indépendamment de l’opérateur de dégradation [12]. Cette contribution s’inscrit dans cette deuxième approche permettant une plus grande flexibilité dans la tâche de reconstruction mais également possédant, sous certaines hypothèses, des garanties théoriques [2, 3].

Les auteurs remercient la Fondation Simone et Cino Del Duca et l’ANR (MEPHISTO, ANR-24-CE23-7039-01) pour avoir financé ces travaux. Ils remercient aussi le Centre Blaise Pascal de l’ENS Lyon pour leur avoir mis à disposition l’infrastructure de calcul SIDUS développé par Emmanuel Quemener [9].

Bien que les algorithmes PnP soient compétitifs, ils possèdent deux limitations majeures ; à savoir qu’ils reposent sur un grand nombre d’itérations avant d’obtenir un résultat satisfaisant et ils sont sensibles à l’initialisation. Pour pallier le problème de rapidité, certains algorithmes comme Deep Plug-and-Play for Image Restoration (DPIR) [13], sortent du cadre PnP en faisant varier le paramètre de bruit du débruiteur en fonction des itérations, en supposant que le niveau de dégradation diminue. Pour résoudre le second problème, une série de travaux se sont intéressés à des approches multiniveaux. Par exemple les auteurs de [1] proposent une méthode multi-échelle qui adopte une perspective bayésienne où des approximations lisses de la log-vraisemblance négative initiale sont considérées. Tang et al. [10], ont quant à eux proposé un schéma de *sketching* multiniveau dans le cadre de l’IRM. Ces approches sont un premier pas vers les approches multiniveaux dans le contexte PnP mais n’exploitent pas le formalisme multiniveau développé en optimisation comme initialement proposé dans [4] pour un schéma variationnel classique basé sur une pénalisation par variation totale (TV). Nous avons récemment proposé dans [5], une extension du formalisme d’optimisation multiniveau au cadre PnP permettant un gain en temps de calcul et en performance par rapport à une approche PnP standard.

**Contribution** – Dans la présente contribution, nous approfondissons l’analyse menée dans [5] afin d’évaluer les performances de notre approche  $ML-PnP$  au vu des choix des opérateurs aux niveaux grossiers. On peut en particulier noter que pour un algorithme multiniveau, une itération est plus coûteuse en nombre d’opérations que sa version simple niveau. Dans l’implémentation de  $ML-PnP$  qui a été proposée, ce surcoût inclut l’évaluation d’un débruiteur coûteux de nombreuses fois [5]. Dans cet article, nous proposons un algorithme  $ML-PnP$ , fondée sur une opération de régularisation sans apprentissage et en forme close rapide à calculer de sorte à ce que le surcoût du multiniveau soit le plus faible possible, tout en gardant la

robustesse à l'initialisation.

**Plan** – La Section 2 présente le cadre général de PnP et ML-PnP. La Section 3 détaille une version accélérée de ML-PnP reposant sur des itérations grossières approchées. Finalement, la Section 4 évalue l'efficacité de la version accélérée par rapport aux approches de l'état de l'art dans un contexte de dématricage d'images.

## 2 Plug-and-Play multiniveau

### 2.1 Plug-and-Play

Nos travaux reposent sur l'algorithme PnP forward-backward, le plus utilisé dans la littérature, qui s'exprime par

$$x_{k+1} = \mathbb{T}(x_k) \quad \text{avec} \quad \mathbb{T} = D_\sigma \circ (\text{Id} - \tau \nabla f) \quad (2)$$

où  $f$  est le terme d'attache aux données (par exemple  $f(x) := \frac{1}{2} \|Ax - z\|^2$ ) et  $D_\sigma : \mathbb{R}^N \rightarrow \mathbb{R}^N$  est un débruiteur, généralement défini par un réseau de neurones. Celui-ci prend en paramètre un niveau de bruit  $\sigma$ , et vérifie la relation  $D_\sigma(\bar{x} + \sigma \varepsilon) \approx \bar{x}$ , où  $\varepsilon \sim \mathcal{N}(0, \text{Id})$ . Ce dernier peut être interprété comme le minimiseur moindre carré (MMSE) suivant

$$D_\sigma := \arg \min_{D_\sigma} \mathbb{E}_{\bar{x}, \varepsilon} \|\bar{x} - D_\sigma(\bar{x} + \sigma \varepsilon)\|_2^2,$$

permettant d'obtenir des garanties sur sa convergence [3]. D'autres résultats de convergence existent, par exemple sous l'hypothèse que le débruiteur est fermement contractant [11].

### 2.2 Plug-and-Play multiniveau

Cette section fournit une description de l'algorithme ML-PnP que nous avons proposé dans [5]. Celui-ci est composé de deux étapes clés : une étape d'initialisation multiniveau, permettant d'obtenir un meilleur point de départ basé sur le choix usuel  $A^\top z$ , et un schéma multiniveau avec une relation de cohérence entre les niveaux permettant d'assurer la stabilité de la solution.

**Notations** – L'algorithme multiniveau opère sur une hiérarchie d'images de différentes résolutions, dont l'indice est représenté par  $j \in \{0, \dots, J-1\}$ , où  $j=0$  représente la *résolution fine* (l'image de haute résolution), et  $j=J-1$  représente la *résolution grossière* (l'image de basse résolution). Pour passer d'une résolution fine à grossière, l'opérateur  $\downarrow$  est utilisé, tandis que le chemin inverse se fait par  $\uparrow$ . De plus,  $\downarrow^j = \downarrow \circ \dots \circ \downarrow$  et  $\uparrow^j = \uparrow \circ \dots \circ \uparrow$  indiquent  $j$  compositions de l'opérateur.

**Motivation du multiniveau** – Le formalisme multiniveau permet une résolution simplifiée à un niveau grossier, dont la solution guidera le niveau plus fin tout en bénéficiant de plus faibles coûts de calculs.

Dans ce cadre, un schéma itératif  $\mathbb{T}_j : \mathbb{R}^{\frac{N}{4^j}} \rightarrow \mathbb{R}^{\frac{N}{4^j}}$  est défini de façon spécifique à chaque niveau  $j$  exprimé par

$$\mathbb{T}_j = D_{j,\sigma} \circ (\text{Id} - \tau \nabla f_j), \quad (3)$$

où  $D_{j,\sigma}$  est un débruiteur appliqué sur une image de  $\frac{N}{4^j}$  pixels et  $\nabla f_j$  est l'attache aux données modélisée en fonction de  $f$ . Dans le cas où  $f_j := \frac{1}{2} \|A_j x - z_j\|^2$  avec  $\|\cdot\|^2$  la norme euclidienne,

$$(\forall x_j \in \mathbb{R}^{\frac{N}{4^j}}) \quad \nabla f_j(x_j) = A_j^\top (A_j x_j - z),$$

où  $A_j = A \circ \uparrow^j$ . En pratique, dans un but de réduction de temps de calcul, ce choix peut légèrement différer. Notons que dans ce contexte, nous pouvons choisir le débruiteur  $D_{j,\sigma}$  en fonction du niveau  $j$ .

**Étape d'initialisation** – De sorte à ne pas être sensible à l'initialisation, ML-PnP commence par appliquer le schéma standard PnP au niveau le plus grossier via  $\mathbb{T}_{J-1}$  jusqu'au niveau qui précède le plus fin, tel que décrit par l'Algorithme 1. Le résultat obtenu en sortie constitue un bon point de départ pour les prochaines itérations au niveau fin, lorsque  $j=0$ . Cependant, pour pleinement bénéficier du cadre multiniveau, les itérations appliquées au niveau fin doivent bénéficier d'une relation de cohérence détaillée dans le paragraphe suivant.

---

#### Algorithme 1 : ML Init

---

```

Choisir  $x \in \mathbb{R}^N$  et  $J \in \mathbb{N}$ 
Fixer  $u_{J-1} = \downarrow^{J-1} x$ 
pour  $j = J-1, \dots, 1$  faire # Niveau grossier vers fin
     $v_j = \mathbb{T}_j \circ \dots \circ \mathbb{T}_j(u_j)$  # PnP au niveau  $j$ 
     $u_{j-1} = \uparrow v_j$  # Transfert au niveau plus fin
fin
retourner  $u_0$ 

```

---

**Schéma multiniveau** – L'algorithme que nous décrivons ici alterne entre une opération multiniveau (ML) et une opération classique PnP. Autrement dit, une telle itération s'exprime comme la composition

$$x_{k+1} = \mathbb{T} \circ \text{ML}(x_k), \quad (4)$$

où  $\mathbb{T}$  est l'opérateur PnP défini à l'équation (2) et ML est un algorithme permettant d'accélérer la reconstruction tout en assurant la stabilité. Pour cela les schémas multiniveaux usuellement employés en optimisation lisse [8] et étendus en optimisation non-lisse [4] requièrent d'imposer une forme de cohérence entre les itérations aux différents niveaux, le problème étant que si le schéma  $\mathbb{T}_j$  est itéré à l'échelle  $j$ , les itérations convergeront vers un point de  $\text{Fix } \mathbb{T}_j = \text{Zer } \mathbb{U}_j$  où  $\text{Fix}$ ,  $\text{Zer}$  renvoient respectivement les points fixes, zéros d'un opérateur et (grâce à la formule de Tweedie)

$$\mathbb{U}_j := \nabla f_j + \sigma^{-2} (\text{Id} - D_{j,\sigma}).$$

Ceci n'est pas désirable, puisqu'on souhaite converger vers un point dans  $\text{Zer } \mathbb{U}_0$ . Pour s'en assurer, dans [5] nous avons proposé de remplacer le schéma  $\mathbb{T}_j$  de l'équation (3) par  $\tilde{\mathbb{T}}_j$  défini comme :

$$\tilde{\mathbb{T}}_j = D_{j,\sigma} \circ (\text{Id} - \tau \nabla f_j - \tau c_j)$$

où  $c_0 = 0$  et pour  $j > 0$  le vecteur  $c_j \in \mathbb{R}^{N/4^j}$  est défini par

$$c_j = \tilde{\mathbb{U}}_{j-1} (\downarrow^{j-1} x_k) - \mathbb{U}_j (\downarrow^j x_k)$$

avec

$$\tilde{\mathbb{U}}_j := \mathbb{U}_j + c_j.$$

Cela assure la cohérence, dans le sens que pour  $j \geq 1$ ,  $\downarrow \circ \tilde{\mathbb{U}}_{j-1} \approx \mathbb{U}_j \circ \downarrow$  dans un voisinage de  $\downarrow^{j-1} x_k$ . Cette relation nous assure que les opérations faites aux niveaux grossiers sont pertinentes par rapport au modèle du niveau fin. Nous choisissons  $c_0 = 0$  car la propriété de cohérence n'est pas nécessaire au niveau le plus fin. L'opération ML obtenue est décrite par l'Algorithme 2.

---

**Algorithme 2** : ML( $x$ )

---

Choisir  $J \in \mathbb{N}$   
 $u_{J-1} = \downarrow^{J-1} x$   
Définir  $(c_0, c_1, \dots, c_{J-1})$   
**pour**  $j = J - 1, \dots, 1$  **faire** # de grossier vers fin  
     $\tilde{T}_j = D_{j,\sigma} \circ (\text{Id} - \tau \nabla f_j - \tau c_j)$  # op. PnP  
     $v_j = \tilde{T}_j \circ \dots \circ \tilde{T}_j(u_j)$  # exécution de PnP au niveau  $j$   
     $u_{j-1} = \downarrow^{j-1} x + \uparrow(v_j - u_j)$  # initialisation à  $j - 1$   
**fin**  
**retourner**  $u_0$

---

### 3 Algorithme multiniveau hybride

Dans [5], nous avons pu montrer que l'algorithme ML-PnP, qui choisit  $D_{j,\sigma} = D_\sigma$  pour tout  $j$  (le débruiteur pouvant opérer sur des images de différentes tailles), permettait d'obtenir de très bonnes performances en terme de reconstruction, stabilité et rapidité.

En revanche, un tel débruiteur peut être coûteux à évaluer, nous proposons donc de le remplacer dans les niveaux grossiers par un opérateur plus rapide à calculer.

**Choix de la régularisation** – Une des régularisations centrales en traitement d'image est la variation totale

$$g(x) := \|Lx\|_{1,2} \quad (5)$$

où  $\|\cdot\|_{1,2}$  est la norme mixte  $\ell_{1,2}$  et  $L$  est l'opérateur de différences finies. Dans [4], où l'approche multiniveau est introduite pour la minimisation de  $f + \lambda g$  dans le cadre non-lisse mais convexe, la cohérence du premier ordre est imposée en ayant recours à une version lissée de  $g$ , plus exactement via son enveloppe de Moreau définie par

$$\sigma g(y) := \min_{x \in \mathbb{R}^N} \frac{1}{2\sigma} \|y - x\|^2 + g(x)$$

et dont le gradient peut être approché par

$$G_\sigma^M := \frac{L^\top}{\sigma} \left( \text{Id} - \text{prox}_{\sigma \|\cdot\|_{1,2}} \right) \circ L,$$

c'est-à-dire que  $\nabla(\sigma g) \approx G_\sigma^M$  [6]. Dans les échelles grossières  $j > 0$ , nous proposons donc d'utiliser  $G_\sigma^M$ , qui sera alors noté  $G_{\sigma,j}^M$ , dans le schéma de débruitage, puisque cet opérateur a une forme close qui se calcule rapidement.

**Intégration dans l'algorithme** – Pour avoir un coût de calcul plus faible dans les niveaux grossiers, quand  $j > 0$  nous considérons  $G_{\sigma,j}^M$  mais ne modifions pas les itérations au niveau le plus fin (i.e., quand  $j = 0$ , nous gardons un débruiteur MMSE) de sorte à avoir une reconstruction compétitive. Ainsi, nous considérons, pour  $j > 0$ ,

$$T_j := \text{Id} - \tau(\nabla f_j + G_{\sigma,j}^M)$$

dans les schémas ML d'initialisation sans cohérence de premier ordre et

$$\tilde{T}_j := \text{Id} - \tau(\nabla f_j + G_{\sigma,j}^M + c_j)$$

pour le schéma multiniveau imposant la cohérence où dans ce cas, la constante  $c_j$  se calcule à partir de

$$U_j := \begin{cases} \nabla f_j + \sigma^{-2}(\text{Id} - D_\sigma^{\text{MMSE}}) & \text{si } j = 0, \\ \nabla f_j + G_{\sigma,j}^M & \text{si } j > 0. \end{cases}$$

L'algorithme résultant de ces itérations approchées au niveau grossier sera nommé ML-PnP Moreau.

## 4 Expériences numériques

**Problème de dématricage** – Ce problème est décrit par un filtre de Bayer présent sur certains capteurs photographiques. Ce filtre est déterministe et est différent pour chaque canal d'une image RGB. En conséquence du filtrage, un motif apparaît sur l'image dégradée, visible sur une grille de  $2 \times 2$  pixels : la composante verte est retenue sur deux pixels, la composante rouge sur un pixel et la composante bleue sur un pixel. Dans les expériences suivantes, nous avons choisi  $\varepsilon \sim \mathcal{N}(0, \sigma^2 \text{Id})$  avec  $\sigma = 0.1$ .

**Méthodes comparées** – Nous comparons notre méthode à celles décrites ci-dessous :

- ML-FB TV : Algorithme forward-backward multiniveau [4] qui minimise  $f + \lambda g$  où  $\lambda > 0$  et la régularisation  $g$  est choisie comme la variation totale en (5).
- PnP : Schéma PnP forward-backward dont les itérations sont décrites par l'équation (2).
- DPIR : Algorithme similaire à PnP, où un débruiteur est intégré dans le schéma HQS (half quadratic splitting), où les paramètres du débruiteur varient en fonction des itérations.

Les paramètres de toutes ces méthodes ainsi que de ML-PnP et ML-PnP Moreau sont les mêmes que ceux choisis dans [5]. Dans les niveaux grossiers des méthodes multiniveaux, nous reprenons la modélisation du dématricage proposée dans [5], c'est-à-dire que pour définir  $A_j$ , le filtre de Bayer est représenté au niveau  $j$  à l'aide de  $\downarrow^j$ .

**Analyse numérique** – Nous comparons ici l'ensemble des algorithmes en mesurant leur efficacité par le PSNR en fonction du temps. La Figure 1 fournit une illustration de l'efficacité des différentes méthodes sur 10 secondes pour les trois choix de  $\bar{x}$  et leur version dégradée  $z$  présentés sur la Figure 2 (figure de gauche). Nous pouvons constater que la méthode ML-PnP Moreau accélère la reconstruction puisque le PSNR est significativement plus élevé que pour toutes les autres méthodes dans les premiers instants et atteint rapidement un plateau. Cependant, la qualité de reconstruction obtenue est un peu plus faible que pour ML-PnP.

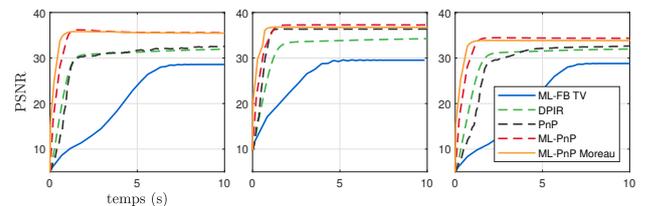


FIGURE 1 : Efficacité de reconstruction des algorithmes multiniveaux et état de l'art mesuré en PSNR en fonction du temps pour le problème de dématricage. Chaque colonne est respectivement associée aux images (A), (B) et (C) présentées sur la Figure 2 (gauche).

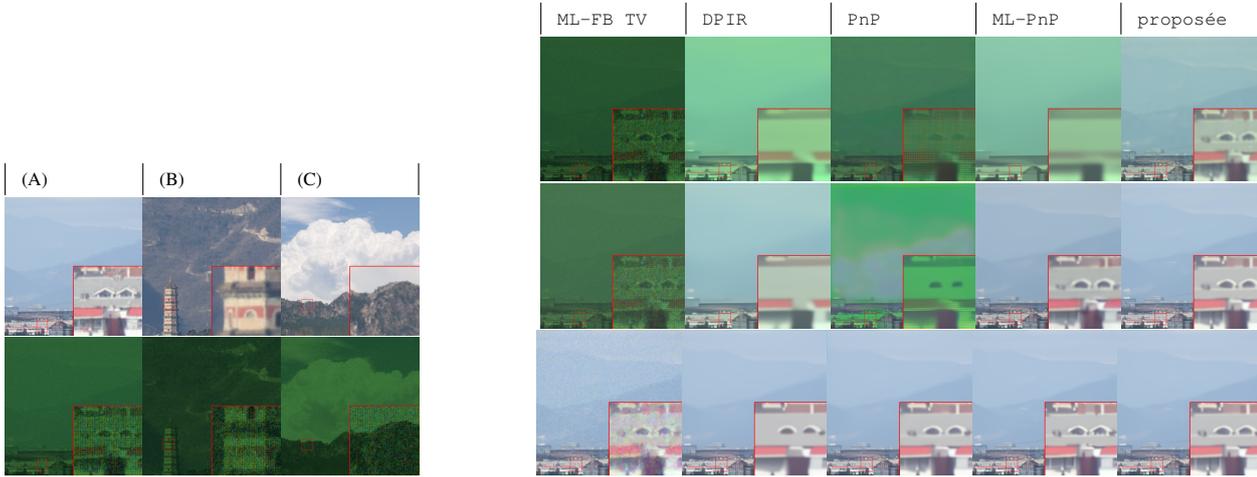


FIGURE 2 : Le tableau d’images à gauche présente trois choix de  $\bar{x}$  (ligne 1) et leur version dégradée  $z$  (ligne 2). Le tableau d’images à droite présente les résultats de reconstruction pour le problème de dématricage sur (A). Chaque colonne présente la reconstruction pour une des méthodes comparées. Les lignes 1, 2 et 3 présentent la reconstruction pour 0.5s, 1s et 20s de calcul respectivement.

Les résultats des reconstructions pour la figure (A) à 0.5s, 1s et 20s sont présentés sur la Figure 2 (droite). Nous remarquons qu’à 0,5s, ML-PnP Moreau fournit des couleurs bien plus proches de  $\bar{x}$  que les autres méthodes et un meilleur niveau de détails. À 1s de calcul, les méthodes ML-PnP et ML-PnP Moreau ont presque convergé, alors que l’effet du dématricage est encore présent pour les autres méthodes.

## 5 Conclusion

Dans cet article, nous avons proposé une variante de l’algorithme multiniveau ML-PnP initialement proposé, en intégrant une régularisation sous forme close dans les échelles grossières, diminuant ainsi les coûts de calculs associés. Nous avons observé que l’algorithme résultant accélère significativement la reconstruction tout en restant compétitif sur la qualité de la reconstruction.

## Références

- [1] J. R. CHAND et M. JACOB : Multi-scale energy (MuSE) framework for inverse problems in imaging. *IEEE Trans. Comput. Imaging*, 10:1250–1265, 2024.
- [2] S. HURAUULT, A. LECLAIRE et N. PAPADAKIS : Gradient step denoiser for convergent plug-and-play. *In International Conference on Learning Representations*, 2022.
- [3] Z. KADKHODAIE et E. P. SIMONCELLI : Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *In Advances in Neural Information Processing Systems*, 6- 14 Dec.2021.
- [4] G. LAUGA, E. RICCIETTI, N. PUSTELNIK et P. GONÇALVES : IML FISTA : A multilevel framework for inexact and inertial forward-backward. application to image restoration. *SIAM J. Imaging Sci.*, 17(3):1347–1376, 2024.
- [5] N. LAURENT, J. TACHELLA, E. RICCIETTI et N. PUSTELNIK : Multilevel Plug-and-Play Image Restoration. hal-050049144, 2025.
- [6] T. D. LUU, J. FADILI et C. CHESNEAU : Sampling from Non-smooth Distributions Through Langevin Diffusion. *Methodology and Computing in Applied Probability*, 23(4):1173–1201, 2021.
- [7] V. MONGA, Y. LI et Y. C. ELДАР : Algorithm unrolling : Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Process. Mag.*, 38(2):18–44, 2021.
- [8] S. G. NASH : A multigrid approach to discretized optimization problems. *Optimization Methods and Software*, 14(1-2):99–116, 2000.
- [9] E. QUEMENER et M. CORVELLEC : SIDUS—the Solution for Extreme Deduplication of an Operating System. *Linux J.*, 2013(235), 2013.
- [10] J. TANG, G. XU, S. MUKHERJEE et C.-B. SCHÖNLIEB : Practical operator sketching framework for accelerating iterative data-driven solutions in inverse problems. *In Proc IEEE Int. Conf. Acoust. Speech Signal. Process.*, Hyderabad, India, 6-11 Avr. 2025.
- [11] M. TERRIS, A. REPETTI, J.-C. PESQUET et Y. WIAUX : Enhanced Convergent PnP algorithms for image restoration. *In Proc IEEE Int. Conf. Image Proc.*, 19-22 Sept. 2021.
- [12] S. V. VENKATAKRISHNAN, C. A. BOUMAN et B. WOHLBERG : Plug-and-Play priors for model based reconstruction. *In IEEE Glob. Conf. Signal Inf. Process. Proc.*, pages 945–948, Austin, TX, USA, 3-5 Dec. 2013.
- [13] K. ZHANG, Y. LI, W. ZUO, L. ZHANG, L. VAN GOOL et R. TIMOFTE : Plug-and-Play image restoration with deep denoiser prior. *IEEE Trans. Pattern Anal. Match. Int.*, 44(10):6360–6376, 2022.
- [14] K. ZHANG, W. ZUO, S. GU et L. ZHANG : Learning deep CNN denoiser prior for image restoration. *In IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21-26 Juil. 2017.