Journal of Mathematical Imaging and Vision (2025) 67:59
https://doi.org/10.1007/s10851-025-01273-7

Plug and Play Splitting Techniques for Poisson Image Restoration

Alessandro Benfenati[1,2]

Received: 25 July 2025 / Accepted: 12 November 2025 / Published online: 25 November 2025
© The Author(s) 2025

Abstract
Plug and Play (PnP) methods achieve remarkable results in the framework of image restoration problems for Gaussian data. Nonetheless, the theory available for the Gaussian case cannot be extended to the Poisson case, due to the non-Lipschitz gradient of the fidelity function, the Kullback-Leibler functional, or the absence of closed-form solution for the proximal operator of such term, leading to employ iterative solvers for the inner subproblem. In this work we extend the idea of PIDSPLIT+ algorithm, exploiting the Alternating Direction Method of Multipliers, to PnP scheme: this allows to provide a closed-form solution for the deblurring step, with no need for iterative solvers. The convergence of the method is assured by employing a firmly non-expansive denoiser. The proposed method, namely PnPSplit$^+$, is tested on different Poisson image restoration problems, showing remarkable performance even in presence of high noise level and severe blurring conditions.

Keywords Image restoration · Poisson data · Plug and Play · Splitting techniques

1 Introduction

Imaging problems arise in several scientific fields, such as Medicine [1, 2], Astronomy [3–5], and Microscopy [6–8]. The mathematical model underlying the physics process is shared among all these disciplines [9], and it reads as

$$\mathbf{g} = \mathcal{N}\left(\mathbf{H}\,\mathbf{x}^\star + b\right), \tag{1}$$

where $\mathbf{x}^\star \in \mathbb{R}^n$ denotes the ground truth image, $\mathbf{H} \in \mathbb{R}^{m\times n}$ is a linear operator perturbing the data, $b \in \mathbb{R}^+$ is a known background parameter, $\mathbf{g} \in \mathbb{R}^m$ is the recorded image, and $\mathcal{N}$ denotes the statistical noise on recorded data. Classical examples of noise model include Gaussian noise [9], Salt&Pepper noise [10], Speckle noise [11], Poisson noise [12], and mixture of Poisson and Gaussian noise [13]. The operator $\mathbf{H}$ is also called Point Spread Function (PSF), since its representation is the registered image of a point source; classical hypotheses on $\mathbf{H}$, abiding by real-life systems properties, are that $\mathbf{H}^\top\mathbf{1} = \mathbf{1}$ and $\sum_{ij} H_{ij} = 1$. The aim of image restoration problems is to recover an estimation of $\mathbf{x}^\star$ given the registered data $\mathbf{g}$ and the operator $\mathbf{H}$. When the recorded data $\mathbf{g}$ are affected by Poisson noise, under a Bayesian framework, that is adopting a *maximum a posteriori* approach [12, 14, 15], one is led to solve the optimization problem

$$\underset{\mathbf{x}\in\mathbb{R}^n_+}{\operatorname{argmin}}\ KL(\mathbf{H}\,\mathbf{x} + b, \mathbf{g}) + \beta\, R(\mathbf{x}), \tag{2}$$

where KL is the generalized Kullback–Leibler functional

$$KL(\mathbf{H}\,\mathbf{x} + b, \mathbf{g}) = \sum_i g_i \log\left(\frac{g_i}{(\mathbf{H}\mathbf{x})_i + b}\right) + (\mathbf{H}\mathbf{x})_i + b - g_i.$$

The operations are intended component-wise, and one assumes $0\log(0) = 0$. In particular, $KL(\cdot, \mathbf{g})$ is a proper, convex, and differentiable functional. The function R is the regularization functional and its role is to preserve the desired characteristics on the estimated solution, such as sharp edges or sparseness, and to control the influence of the noise on the estimated solution. Common choices for R include proper, lower semi-continuous (l.s.c.) convex function, such as ℓ_2 regularization, which goes also under the name Tikhonov regularization [16, 17] or Ridge Regression in other frameworks [18], ℓ_1 norm for promoting sparsity on the solution [19], a convex combination of ℓ_2 and ℓ_1 norms, commonly referenced to as elastic net [20]. Another popular choice is the total variation functional [21], for promoting sharp edges,

✉ Alessandro Benfenati
alessandro.benfenati@unimi.it

[1] Environmental and Science Policy Department, University of Milan, Via Celoria 2, 20133 Milan, Italy

[2] Gruppo Nazionale Calcolo Scientifico, INDAM, Piazzale Aldo Moro, 5, 00185 Rome, Italy

and its offsprings [22, 23]. The parameter $\beta \in \mathbb{R}^+$ is the regularization parameter and balances the trade-off between the KL and R. A common requirement in imaging problems is that the solution components are non-negative, since they represent pixels' intensity: therefore, the estimated solution is required to belong to the non-negative orthant $\mathbb{R}^n_+$.

The literature presents a plethora of variational methods to solve this particular instance of restoration problem: among them one can find gradient approaches [24] and the related variation [25–27], Bregman iterative methods [28–30], proximal approaches [31]. The Alternating Direction Method of Multipliers (ADMM) has gained a predominant role in image restoration problem [32–34], showing particularly interesting results in managing optimization problems with linear constraints.

The seminal work [35] introduced a novel approach, called Plug and Play (PnP) technique. This strategy consists of solving optimization problems, whose objective functional encompasses two terms. Employing splitting techniques as ADMM, the authors in [35] observed that the update for one of the variables reads actually as a Gaussian denoising step: therefore, they propose to substitute such updating step with an off-the-shelf denoiser D, such as Block-Matching and 3D Filtering (BM3D) [36], Nonlocal Mean Filter (NLM) [37]. Modern approaches encompass also the usage of deep neural networks, tailored for Gaussian denoising [38].

The main hypothesis is that such denoiser is the proximal operator of some function R: the numerical experience showed the remarkable results of this approach. The research interest then moved to investigate the theoretical hypothesis to have on the denoiser for assuring the convergence of PnP: indeed, fixed point theory tells us that such denoiser needs to be firmly non-expansive [39, 40], but unfortunately most of the employed denoisers do not fulfill this requirement [41], despite their impressive performance results. Even classical neural networks, that show remarkable performances in Gaussian denoising tasks, cannot satisfy this requirement, unless properly trained with tailored loss function [42]. The scientific research explored the control of the Lipschitz constant of the neural network [43–45], but the quality of such control is not strong enough to ensure the convergence property and moreover the computational cost is rather high. In [46] the PnP framework has been addressed by considering it as a constrained problem under an ADMM approach, where a discrepancy principle is used in reformulating the problem. This approach allows to automatically chose the regularization parameter. Different techniques have been explored to assure convergence of PnP method: bounded denoisers assure fixed point convergence [47]; in [48] an incremental version of PnP with explicit requirements on the denoiser, namely its firmly non-expansiveness, assures the convergence while maintaining scalability in terms of speed and memory. In [49] under the hypotheses of the denoiser being averaged and the convexity of the data fidelity term the PnP scheme converges, and moreover it is shown that some of the employed denoisers are indeed the proximal operator of particular functions, *e.g.*, the NLM is the prox of a quadratic convex function.

One has to mention alternative approaches to PnP, which try to address the theoretical issues posed by PnP. The Regularization by Denoising (RED) method [50] is among them; it tries to overcome the PnP limitations by requiring the denoiser to have a symmetric Jacobian and to be locally homogeneous: unfortunately, although the theoretical framework is very rich and interesting, the majority of the employed denoiser do not satisfy this requirements. RED has been then investigated from different points of view: it has been reformulated [51] as a constrained optimization problem (RED-PRO), where the least square minimum is projected on the fixed point sets of demicontractive denoisers, which reveal to be convex sets. In [52] the RED-PRO has been reversed following a discrepancy principle, leading to a constrained RED approach (CRED): the RED functional is minimized under the discrepancy between the recovered solution and the data $\mathbf{g}$. Deep equilibrium models have been recently studied for addressing Poisson image restoration [53].

A further step was done considering Gradient Step Denoisers [54], where the denoising step is carried out by subtracting to the current image the gradient of a parametrized function g_ϑ : a classical and performant choice is $g_\vartheta(\mathbf{x}) = 1/2\|\mathbf{x} - n(\mathbf{x})\|_2^2$, where n is a denoising neural network. This particular strategy allows for a more solid theoretical convergence property, and from the practical point of view, it is possible to learn the denoiser without compromising the numerical performance.

Most of the previous research on PnP methods focused on data corrupted by Gaussian noise. Image corrupted by Poisson noise presents different challenges, mainly for the presence of the Kullback–Leibler divergence as part of the objective functional. The seminal work [55] adopts a Bregman approach for designing a tailored method for deblurring and denoising tasks in presence of Poisson noise: the remarkable numerical results are supported by solid theoretical result. Adopting a different strategy, in [56] a novel denoiser is created for Poisson data employing a denoiser based on Schroedinger equation's solution from quantum physics. An ADMM approach is adopted in [57], showing reliable results also in presence of high-level Poisson noise. Besides variational methods, the authors in [58] explore Bayesian approaches, in particular Langevin approaches, for addressing image restoration for Poisson data.

The variational methods previously mentioned show remarkable results in terms of reconstruction, both in denoising and deblurring tasks, and rely on solid theoretical basis. Nonetheless, all of them rely on iterative methods for solving

the deblurring step, meaning that either one has to accept an inexact solution to the inner problem or wait for the convergence of the inner iterative procedure. In this work, instead, the split Bregman approach presented in [59] is exploited, *i.e.*, coupling it with the PnP idea of substituting the proximity operator with an off-the-shelf denoiser, chosen to satisfy the firmly non-expansive property. The split Bregman technique allows to avoid the usage of iterative methods for the deblurring step, by solving a trivial least square minimization problem, which possesses the nice property of having an unique solution. This significantly reduces the computational cost and, indirectly, the computational time. When the chosen denoiser satisfies the firmly non-expansiveness hypothesis, one can extend the theoretical result of [59] for proving the convergence of the proposed scheme. The proposed method is then compared with state-of-the-art algorithms and tested under different blurring conditions and Poisson noise levels.

This work is organized as follows. Section 2 initially provides a background on ADMM and on Plug and Play methods, providing convergence results for the former and setting the notations used throughout the work. Section 3 presents the proposed method, providing the convergence result. Section 4 assesses the performance of the proposed method, comparing it with state-of-the-art algorithms, testing under extreme perturbation conditions and under different blurring operator and, finally, employing a denoiser which does not satisfy the theoretical requirements for convergence. Eventually, Sect. 5 draws the final considerations and consider possible future extensions of this work.

Notation. The set $\mathbb{R}^n$ denotes the real vector space of dimension n; $\mathbb{R}^{m\times n}$ denotes real matrices with m rows and n columns. Bold capital symbols ($\mathbf{A}$, $\mathbf{\Omega}$, ...) denote matrices; bold small symbols ($\mathbf{x}$, $\boldsymbol{\lambda}$, ...) denote vectors. For a vector $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} \geq \mathbf{0}$ means that each element of $\mathbf{x}$ is greater or equal to zero. The set $\mathbb{R}^n_+$ is the non-negative orthant of $\mathbb{R}^n$: $\mathbb{R}^n_+ = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x} \geq 0\}$. Italic and Greek letters denote scalars in $\mathbb{R}$. $\|\cdot\|_p$ stands for the ℓ_p norm. proj_A denotes the projection onto the set A. The set Γ_0 denotes the set of convex, proper and lower semi-continuous (l.s.c.) functions. The indicator function of a set C is denoted with $\iota_C(\mathbf{x})$, where

$$\iota_C(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in C \\ +\infty & \text{otherwise} \end{cases}$$

The proximity operator (called also proximal operator or just prox) of a function f at a point $\mathbf{c}$ is denoted with $\text{prox}_f(\mathbf{c})$, and it consists of

$$\text{prox}_f(\mathbf{c}) = \underset{\mathbf{x}}{\text{argmin}} \; f(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{c}\|_2^2.$$

2 Plug and Play Methods

Splitting methods address a general minimization problem of the form

$$\underset{\mathbf{x}}{\text{argmin}} \; \psi(\mathbf{x}) + \beta\varphi(\mathbf{M}\mathbf{x}) \tag{3}$$

where $\varphi, \psi \in \Gamma_0$, with φ being differentiable, and $\mathbf{M}$ is a linear operator. Note that Problem (2) can be cast in this form by setting $\psi \equiv \beta R$, $\varphi \equiv \beta^{-1} KL$ and $\mathbf{M} = \mathbf{H}$. Introducing $\mathbf{M}\mathbf{x} = \mathbf{w}$, the problem can be recast as

$$\underset{\mathbf{x},\mathbf{w}}{\text{argmin}} \; \psi(\mathbf{x}) + \beta\varphi(\mathbf{w}), \quad \text{such that } \mathbf{M}\mathbf{x} = \mathbf{w}.$$

The new constraint can be embedded in the objective functional, leading to the augmented Lagrangian:

$$\mathcal{L}(\mathbf{x}, \mathbf{w}, \boldsymbol{\lambda}) = \psi(\mathbf{x}) + \beta\varphi(\mathbf{w}) + \frac{1}{2\gamma}\|\mathbf{M}\mathbf{x} - \mathbf{w} + \boldsymbol{\lambda}\|_2^2 - \frac{1}{2\gamma}\|\boldsymbol{\lambda}\|_2^2,$$

where the substitution $\boldsymbol{\lambda} \leftarrow \gamma\boldsymbol{\lambda}$ has been made, with a slight abuse of notation. This leads to solve the saddle point problem

$$\underset{\mathbf{x},\mathbf{w}}{\text{argmin}} \; \underset{\boldsymbol{\lambda}}{\text{argmax}} \; \mathcal{L}(\mathbf{x}, \mathbf{w}, \boldsymbol{\lambda}). \tag{4}$$

The popular Alternating Direction Method of Multipliers (ADMM) [33] depicted in Algorithm 1 allows solving (4) under suitable hypotheses.

Algorithm 1 ADMM

Set $\mathbf{x}^0$, $\mathbf{w}^0$ and $\boldsymbol{\lambda}^0$ accordingly, select the parameter $\gamma > 0$.
for $k = 0, 1, \ldots$ **do**

$$\mathbf{x}^{k+1} = \underset{\mathbf{x}\geq\mathbf{0}}{\text{argmin}} \; \psi(\mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{M}\mathbf{x} - \mathbf{w}^k + \boldsymbol{\lambda}^k\|_2^2$$

$$\mathbf{w}^{k+1} = \underset{\mathbf{w}}{\text{argmin}} \; \beta\varphi(\mathbf{w}) + \frac{1}{2\gamma}\|\mathbf{M}\mathbf{x}^{k+1} - \mathbf{w} + \boldsymbol{\lambda}^k\|_2^2$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \mathbf{M}\mathbf{x}^{k+1} - \mathbf{w}^{k+1}$$

end for

Remark 1 The update step of the variable $\mathbf{w}$ consists of the proximity operator of the function φ computed at $\mathbf{M}\mathbf{x}^{k+1} + \boldsymbol{\lambda}^k$.

The following result [59, Proposition 2.2] provides the convergence results for the sequences $\{\boldsymbol{\lambda}^k\}_k$ and $\{\mathbf{w}^k\}_k$ and assesses the requirements to be met for having the iterates $\{\mathbf{x}^k\}_k$ solve the primal problem (3).

Proposition 1 *([59]) For any starting point and for any $\gamma \in \mathbb{R}^+$ the sequences $\{\boldsymbol{\lambda}^k\}_k$ and $\{\boldsymbol{w}^k\}_k$ generated by Algorithm 1 converge. The sequence $\{\boldsymbol{x}^k\}_k$ calculated by Algorithm 1 converges to a solution of the primal problem (3) if one of the following conditions is met:*

1. *The primal problem has one and only one solution*
2. *The optimization problem*

$$\operatorname*{argmin}_{\boldsymbol{x}} \ \psi(\boldsymbol{x}) + \frac{1}{2\gamma}\|\boldsymbol{M}\boldsymbol{x} - \hat{\boldsymbol{w}} + \hat{\boldsymbol{\lambda}}\|_2^2$$

has a unique solution, where

$$\hat{\boldsymbol{w}} = \lim_{k\to\infty} \boldsymbol{w}^k, \quad \hat{\boldsymbol{\lambda}} = \lim_{k\to\infty} \boldsymbol{\lambda}^k$$

The seminal work [35] observed that the update rule for $\mathbf{w}$ in Algorithm 1 can be interpreted as a Gaussian denoising step on the variable $\mathbf{w}$, with a regularization function φ. Therefore, they proposed to plug in an off-the-shelf Gaussian denoiser $\mathcal{D}_{\gamma\beta}$ instead of the proximal step, where $\gamma\beta$ is the standard deviation of the Gaussian noise to be removed. The method takes the name of Plug and Play (PnP), and it is depicted, in its general formulation, in Algorithm 2. Some examples for the denoiser used in PnP schemes are BM3D [36] or Nonlocal Mean Filter [37], or trained deep neural networks [42]. The advantage of this strategy is twofold: one does not need to select a priori a regularization function φ, and furthermore, once chosen, one can avoid to compute the proximal operator of φ, via a direct formula-as in the ℓ_1 case-or via an iterative method, *e.g.*, when φ is the total variation regularization.

Algorithm 2 Plug and Play

Set $\mathbf{x}^0$, $\mathbf{w}^0$ and $\boldsymbol{\lambda}^0$ accordingly, select the parameter $\gamma > 0$.
for $k = 0, 1, \ldots$ **do**

$$\mathbf{x}^{k+1} = \operatorname*{argmin}_{\mathbf{x}\geq\mathbf{0}} \ \psi(\mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{M}\mathbf{x} - \mathbf{w}^k + \boldsymbol{\lambda}^k\|_2^2$$

$$\mathbf{w}^{k+1} = \mathcal{D}_{\gamma\beta}(\mathbf{M}\mathbf{x}^{k+1} + \boldsymbol{\lambda}^k)$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \mathbf{M}\mathbf{x}^{k+1} - \mathbf{w}^{k+1}$$

end for

This strategy proved to achieve remarkable results in terms of reconstruction quality and computational time: the numerical experience [35, 60, 61] showed that this method is able to exploit both the properties of the original variational model and the noise removal abilities of the chosen denoiser.

Nonetheless, such an approach does not come without presenting several challenges. Consider the case in which the chosen denoiser is the prox operator of an *unknown* function: this amounts to *implicitly* defining the primal problem to solve and the relative objective function, the latter consisting of the data fidelity ψ (selected according to the noise affecting the data) and of a regularization function ρ such that $\mathcal{D}(\cdot) = \operatorname{prox}_{\rho}(\cdot)$. The implicit optimization problem is therefore

$$\operatorname*{argmin}_{\mathbf{x}} \ \psi(\mathbf{x}) + \rho(\mathbf{M}\mathbf{x}).$$

For example, if the denoiser is the soft thresholding operator, then it is a common knowledge that the function ρ is the ℓ_1 norm; it can be proven that a class of linear denoisers, such as kernel and symmetric denoisers, can be expressed as proximity operators of some convex functions [62].

On the other hand, if the chosen denoiser $\mathcal{D}$ does not meet the requirements to be the proximal operator of a function, then the convergence of PnP method is no longer related to the primal problem; the focus is on consensus equilibrium formulation: this amounts to have

$$\tilde{\mathbf{x}} = \operatorname{prox}_{\psi}(\tilde{\mathbf{x}} - \tilde{\boldsymbol{\lambda}}), \quad \text{and} \quad \tilde{\mathbf{x}} = \mathcal{D}(\tilde{\mathbf{x}} + \tilde{\boldsymbol{\lambda}}),$$

where $\tilde{\mathbf{x}}$ is the restored solution and $\tilde{\boldsymbol{\lambda}}$ can be interpreted as the noise component removed from the data. For a more thoughtful discussion, see [63].

This work focuses on the latter case: select and/or train a denoiser $\mathcal{D}$ which is a firmly non-expansive operator, and it is therefore the resolvent of a maximally monotone operator [39, 40], aiming to exploit classical convergence results on ADMM.

3 Proposed Method

In the Poisson case, PnP methods typically require an iterative solver for the inner deblurring subproblem, see for example [55–57]. This work exploits a clever strategy originally introduced in [32, 59], which allows one to avoid such iterative procedures. Such strategy is then combined with the PnP approach, and by carefully choosing the denoiser, the convergence behavior of the proposed method is ensured.

3.1 Previous Work: PIDSPLIT+

The authors in [59] generalized the method proposed in [32], using a common but clever mathematical trick: adding 0 to the objective functional, which in this case amounts to the scalar product of $\mathbf{x}$ and the zero vector. The optimization problem (2) is slightly modified by adding the term $\langle \mathbf{x}, \mathbf{0}\rangle$ and by introducing the indicator function $\iota_{\mathbb{R}^n_+}$:

$$\operatorname*{argmin}_{\mathbf{x}} \ \langle \mathbf{x}, \mathbf{0}\rangle + KL(\mathbf{H}\mathbf{x} + b, \mathbf{g}) + \beta\, R(\mathbf{x}) + \iota_{\mathbb{R}^n_+}(\mathbf{x}). \tag{5}$$

Introducing the matrix $\mathbf{M} = \left(\mathbf{H}^\top, \mathbf{I}_\mathrm{d}, \mathbf{I}_\mathrm{d}\right)^\top$ the problem (5) can be restated as

$$\operatorname*{argmin}_{\mathbf{x},\mathbf{w}} \ \langle \mathbf{x}, \mathbf{0}\rangle + \varphi(\mathbf{w}), \quad \text{s.t.} \quad \mathbf{M}\mathbf{x} = \mathbf{w} \Leftrightarrow \begin{pmatrix} \mathbf{H} \\ \mathbf{I}_\mathrm{d} \\ \mathbf{I}_\mathrm{d} \end{pmatrix} \mathbf{x} = \begin{pmatrix} \mathbf{w}_1 - b \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{pmatrix}$$

which abides to the formulation in (3) with

$$\varphi(\mathbf{w}) = KL(\mathbf{w}_1, \mathbf{g}) + \beta\, R(\mathbf{w}_2) + \iota_{\mathbb{R}^n_+}(\mathbf{w}_3), \quad \psi(\mathbf{x}) = \langle \mathbf{x}, \mathbf{0} \rangle .$$

This can be easily generalized when the regularization function R encompasses a linear operator $\mathbf{L}$, as $R(\mathbf{Lx})$: the matrix $\mathbf{M}$ reads hence as $\mathbf{M}^\top = \left(\mathbf{H}^\top, \mathbf{L}^\top, \mathbf{I}_\mathrm{d}\right)^\top$. The natural next step is to apply Algorithm 1 to this problem. In particular, the update step for $\mathbf{x}^{k+1}$ reads as

$$\mathbf{x}^{k+1} = \operatorname*{argmin}_{\mathbf{x}} \ \langle \mathbf{x}, \mathbf{0} \rangle + \frac{1}{2\gamma} \|\mathbf{Mx} - \mathbf{w}^k + \boldsymbol{\lambda}^k\|_2^2$$

which amounts to solving

$$\left(\mathbf{H}^\top \mathbf{H} + 2\mathbf{I}_\mathrm{d}\right) \mathbf{x} = \mathbf{H}^\top \left(\mathbf{w}^k - \boldsymbol{\lambda}^k\right)$$

The system matrix is square and non singular: therefore, it has one and only one solution: this leads to satisfy condition ii) of Proposition 1, and therefore the whole method converges. Moreover, assuming the usual hypotheses on the PSF $\mathbf{H}$, the solution of such system can be easily computed by means of FFT.

Due to the separability of the components of the vector $\mathbf{w}$, the update for $\mathbf{w}^{k+1}$ is straightforward:

- The component $\mathbf{w}_1^k$ is computed as the proximal operator of the Kullback–Leibler functional $KL(\cdot, \mathbf{g})$:

$$\begin{aligned} \mathbf{w}_1^{k+1} &= \operatorname*{argmin}_{\mathbf{w}_1} \ KL(\mathbf{w}_1, \mathbf{g}) + \frac{1}{2\gamma} \|\mathbf{H}\,\mathbf{x}^{k+1} - \mathbf{w}_1 + \boldsymbol{\lambda}_1^k\|_2^2 \\ &= \operatorname{prox}_{\gamma\, KL(\cdot + b, \mathbf{g})} \left(\mathbf{H}\mathbf{x}^{k+1} + \boldsymbol{\lambda}_1^k\right) \\ &= \frac{1}{2} \Bigg(\mathbf{H}\mathbf{x}^{k+1} + b + \boldsymbol{\lambda}_1^k - \gamma \\ &\quad + \sqrt{\left(\mathbf{H}\mathbf{x}^{k+1} + b + \boldsymbol{\lambda}_1^k - \gamma\right)^2 + 4\gamma\, \mathbf{g}} \Bigg), \end{aligned}$$

  where the operations are component-wise.
- The component $\mathbf{w}_2$ is given by the proximity operator of the regularization function:

$$\begin{aligned} \mathbf{w}_2^{k+1} &= \operatorname*{argmin}_{\mathbf{w}_2} \ \beta\, R(\mathbf{w}_2) + \frac{1}{2\gamma} \|\mathbf{x}^{k+1} - \mathbf{w}_2 + \boldsymbol{\lambda}_2^k\|_2^2 \\ &= \operatorname{prox}_{\beta\gamma R} \left(\mathbf{x}^{k+1} + \boldsymbol{\lambda}_2^k\right) \end{aligned}$$

- The third element of $\mathbf{w}$ is the projection on the non-negative orthant:

$$\begin{aligned} \mathbf{w}_3^{k+1} &= \operatorname*{argmin}_{\mathbf{w}_3} \ \iota_{\mathbb{R}^n_+}(\mathbf{w}_3) + \frac{1}{2\gamma} \|\mathbf{x}^{k+1} - \mathbf{w}_3 + \boldsymbol{\lambda}_3^k\|_2^2 \\ &= \operatorname{proj}_{\mathbb{R}^n_+} \left(\mathbf{x}^{k+1} + \boldsymbol{\lambda}_3^k\right) \end{aligned}$$

These steps are gathered in Algorithm 3, together with the final updates of the Lagrangian multipliers (which are not listed one by one for the sake of brevity).

Algorithm 3 PIDSPLIT+[59]

Set $\mathbf{x}^0$, $\mathbf{w}^0$ and accordingly $\boldsymbol{\lambda}^0$; select the parameter $\gamma > 0$.
for $k = 0, 1, \ldots$ **do**

$$\mathbf{x}^{k+1} = \left(\mathbf{H}^\top \mathbf{H} + 2\mathbf{I}_\mathrm{d}\right)^{-1} \left[\mathbf{H}^\top \left(\mathbf{w}_1^k - \boldsymbol{\lambda}_1^k\right) + \mathbf{w}_2^k - \boldsymbol{\lambda}_2^k + \mathbf{w}_3^k - \boldsymbol{\lambda}_3^k\right]$$

$$\mathbf{w}_1^{k+1} = \frac{1}{2} \left(\mathbf{H}\mathbf{x}^{k+1} + b + \boldsymbol{\lambda}_1^k - \gamma + \sqrt{\left(\mathbf{H}\mathbf{x}^{k+1} + b + \boldsymbol{\lambda}_1^k - \gamma\right)^2 + 4\gamma\, \mathbf{g}}\right)$$

$$\mathbf{w}_2^{k+1} = \operatorname{prox}_{\gamma R} \left(\mathbf{x}^{k+1} + \boldsymbol{\lambda}_2^k\right)$$

$$\mathbf{w}_3^{k+1} = \operatorname{proj}_{\mathbb{R}^n_+} \left(\mathbf{x}^{k+1} + \boldsymbol{\lambda}_3^k\right)$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \mathbf{M}\mathbf{x}^{k+1} - \mathbf{w}^{k+1}$$

end for

3.2 PnPSplit$^+$

With the aim of adopting a PnP approach, the update rule for $\mathbf{w}_2$ takes again the form of a Gaussian denoising step: therefore, following the original PnP idea, one employs a Gaussian denoiser $\mathcal{D}_{\beta\gamma}$ in place of the proximal operator of R. This choice leads to a novel version of this splitting algorithm, called PnPSplit$^+$, which exploits the splitting idea of [59] and the possibility to select an off-the-shelf denoiser, instead of meticulously selecting a regularization function R and devising tailored algorithm for computing its proximity operator.

Algorithm 4 PnPSplit$^+$

Set $\mathbf{x}^0$, $\mathbf{w}^0$ and accordingly $\boldsymbol{\lambda}^0$; select the parameter $\gamma > 0$.
for $k = 0, 1, \ldots$ **do**

$$\mathbf{x}^{k+1} = \left(\mathbf{H}^\top \mathbf{H} + 2\mathbf{I}_\mathrm{d}\right)^{-1} \left[\mathbf{H}^\top \left(\mathbf{w}_1^k - \boldsymbol{\lambda}_1^k\right) + \mathbf{w}_2^k - \boldsymbol{\lambda}_2^k + \mathbf{w}_3^k - \boldsymbol{\lambda}_3^k\right]$$

$$\mathbf{w}_1^{k+1} = \frac{1}{2} \left(\mathbf{H}\mathbf{x}^{k+1} + b + \boldsymbol{\lambda}_1^k - \gamma + \sqrt{\left(\mathbf{H}\mathbf{x}^{k+1} + b + \boldsymbol{\lambda}_1^k - \gamma\right)^2 + 4\gamma\, \mathbf{g}}\right),$$

$$\mathbf{w}_2^{k+1} = \mathcal{D}_{\beta\gamma} \left(\mathbf{x}^{k+1} + \boldsymbol{\lambda}_2^k\right)$$

$$\mathbf{w}_3^{k+1} = \operatorname{proj}_{\mathbb{R}^n_+} \left(\mathbf{x}^{k+1} + \boldsymbol{\lambda}_3^k\right)$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \mathbf{M}\mathbf{x}^{k+1} - \mathbf{w}^{k+1}$$

end for

The main advantage of this approach is that the deblurring step is computed with a direct explicit formula, without relying on an iterative solver, reducing significantly the computational cost and time.

The denoiser, however, should be properly trained (or selected) in order to ensure the convergence behavior of PnPSplit$^+$ algorithm: this requires that such denoiser is firmly non-expansive [40], as already discussed in Sect. 2. If the

selected denoiser is a convolutional neural network, the latter network can be trained in order to satisfy this requirement, as presented in [42]. The strategy to train such a denoiser is briefly recalled below.

Consider the differential operator $Q_{\vartheta} = 2D_{\vartheta} - \mathbf{I}_{\mathrm{d}}$, where ϑ are the trainable parameters: classical results state that the denoiser D_{ϑ} is firmly non-expansive if and only if Q_{ϑ} is non-expansive: therefore the training of D_{ϑ} should be carried out by solving

$$\operatorname*{argmin}_{\vartheta} \sum_i L(D_{\vartheta}(\mathbf{x}_i), \mathbf{y}_i) \quad \text{such that } Q_{\vartheta} \text{ is non-expansive,}$$

where $\{\mathbf{x}_i, \mathbf{y}_i\}_i$ is the dataset of noisy and clean images for the training and L is the loss function, usually MSE score, used for training. The authors in [42] assume that Q_{ϑ} is differentiable for any ϑ; therefore the requirement for the non-expansiveness amounts to

$$\|\nabla Q_{\vartheta}(\mathbf{x})\| \leq 1 \quad \forall \mathbf{x}.$$

Unfortunately, this cannot be met for each $\mathbf{x}$; hence, in [42] this constraint is imposed on every line $[\mathbf{x}_i, D_{\vartheta}(\mathbf{x}_i)]$, *i.e.*, on each point of the form $\tilde{\mathbf{x}}_i = \delta_i \mathbf{x}_i + (1-\delta_i) D_{\vartheta}(\mathbf{x}_i)$, with δ_i randomly drawn from an uniform distribution on the interval $[0, 1]$. The training phase for the denoiser reads hence as

$$\operatorname*{argmin}_{\vartheta} \sum_i L(D_{\vartheta}(\mathbf{x}_i), \mathbf{y}_i) + \lambda \max\{\|\nabla Q_{\vartheta}(\tilde{\mathbf{x}}_i)\|^2, 1-\varepsilon\}, \tag{6}$$

where λ is a nonnegative regularization parameter and $\varepsilon \in (0, 1)$ allows to control the constraints. The requirement on D_{ϑ} to be differentiable can be overcome: automatic differentiation, the standard technique used in neural network training, allows one to consider denoisers implementing non-smooth activation function such as ReLU (see [42, Remark 3.3] and [64–67] for more theoretical insights).

Remark 2 The training in (6) relies on computing the spectral norm $\|\nabla Q_{\vartheta}(\mathbf{x})\|$ for an image $\mathbf{x}$: as explained in [42, Remark 3.4] this is accomplished by using the power iterative method and backpropagation for computing the Jacobian.

The convergence result for Algorithm 4 follows from Proposition 1, considering the further requirement on the denoiser.

Proposition 2 *Let D_{ϑ} a firmly non-expansive Gaussian denoiser, that is the resolvent of a Maximally Monotone Operator A. Set $\mathcal{D}_{\gamma\beta} = D_{\vartheta}$ in Algorithm 4. For any $\boldsymbol{x}^0, \boldsymbol{w}^0$ and for any $\gamma \in \mathbb{R}^+$ the sequences $\{\boldsymbol{\lambda}^k\}_k$ and $\{\boldsymbol{w}^k\}_k$ generated by PnPSplit$^+$converge. The sequence $\{\boldsymbol{x}^k\}_k$ computed in PnPSplit$^+$converges to $\tilde{\boldsymbol{x}}$ such that*

$$0 \in \boldsymbol{H}^{\top} \nabla KL(\boldsymbol{H}\tilde{\boldsymbol{x}} + b; \boldsymbol{g}) + A(\tilde{\boldsymbol{x}}) + N_{\mathbb{R}^n_+}(\tilde{\boldsymbol{x}}),$$

where $N_{\mathbb{R}^n_+}$ is the normal cone to $\mathbb{R}^n_+$, if one of the following conditions is met:

i) The primal problem has one and only one solution
ii) The optimization problem

$$\operatorname*{argmin}_{\boldsymbol{x}} \psi(\boldsymbol{x}) + \frac{1}{2\gamma}\|\boldsymbol{M}\boldsymbol{x} - \hat{\boldsymbol{w}} + \hat{\boldsymbol{\lambda}}\|_2^2, \quad \hat{\boldsymbol{w}} = \lim_{k\to\infty} \boldsymbol{w}^k, \quad \hat{\boldsymbol{\lambda}} = \lim_{k\to\infty} \boldsymbol{\lambda}^k \tag{7}$$

has an unique solution.

Proof Since D_{ϑ} is firmly non-expansive, the update step in $\mathbf{w}_2$ amounts to compute the resolvent operator of a maximally monotone operator A:

$$\mathbf{w}_2^{k+1} = (A + \mathbf{I}_{\mathrm{d}})^{-1}(\mathbf{x}^{k+1} + \boldsymbol{\lambda}_2^k).$$

The updates for $\mathbf{w}_1^{k+1}$ and $\mathbf{w}_3^{k+1}$ are the proximal operators of the KL and the projection on the non-negative orthant, respectively: therefore, the whole update for $\mathbf{w}^{k+1}$ allows recasting the convergence proof of PnPSplit$^+$into the classical one of the ADMM. Moreover, from the optimality condition one has

$$\begin{aligned}
0 &= \mathbf{M}^{\top}\left(\mathbf{M}\tilde{\mathbf{x}} - \tilde{\mathbf{w}} + \tilde{\boldsymbol{\lambda}}\right)\\
0 &\in \nabla KL(\mathbf{w}_1; \mathbf{g}) + A(\mathbf{w}_2) + N_{\mathbb{R}^n_+}(\mathbf{w}_3) - \frac{1}{\gamma}(\mathbf{M}\tilde{\mathbf{x}} - \tilde{\mathbf{w}} + \boldsymbol{\lambda})\\
0 &= \mathbf{M}\tilde{\mathbf{x}} - \tilde{\mathbf{w}}
\end{aligned}$$

then, plugging in the constraint $0 = \mathbf{M}\tilde{\mathbf{x}} - \tilde{\mathbf{w}}$

$$\begin{aligned}
0 &= \mathbf{M}^{\top}\tilde{\boldsymbol{\lambda}}\\
0 &\in \nabla KL(\mathbf{H}\tilde{\mathbf{x}} + b; \mathbf{g}) + A(\tilde{\mathbf{x}}) + N_{\mathbb{R}^n_+}(\tilde{\mathbf{x}}) - \gamma^{-1}\tilde{\boldsymbol{\lambda}},
\end{aligned}$$

leading thus to

$$0 \in \mathbf{H}^{\top} \nabla KL(\mathbf{H}\tilde{\mathbf{x}} + b; \mathbf{g}) + A(\tilde{\mathbf{x}}) + N_{\mathbb{R}^n_+}(\tilde{\mathbf{x}}).$$

Moreover, as already stated for the PIDSplit+ algorithm, the update step for $\mathbf{x}^{k+1}$ consists of solving a square linear system whose matrix is non singular; therefore, the solution is unique: this amounts to satisfy item *ii)*. □

Remark 3 The proof of Proposition 2 regards the case of firmly non-expansive denoisers. If one manages to employ a D_{ϑ} such that it is an actual proximal operator, *i.e.*, $D_{\vartheta} = \mathrm{prox}_{\rho}$ for some unknown ρ, then the convergence follows directly from Proposition 2. In this case, however, the limit point $\tilde{\mathbf{x}}$ solves a primal problem whose objective functional is

unknown, so that the objective functional of the corresponding primal problem is $KL(\mathbf{H}\cdot;\mathbf{g}) + \rho(\cdot) + \iota_{\mathbb{R}^n_+}$.

Remark 4 The performance of ADMM is related to the choice of the parameter γ: the literature [68–70] presents an adaptive strategy to overcome this issue. Such strategy relies on two quantities, namely the primal and dual residuals:

$$\begin{aligned} \mathbf{p}^k &= \mathbf{M}\mathbf{x}^k - \mathbf{w}^k - b \\ \mathbf{s}^k &= \frac{1}{\gamma}\mathbf{M}^t\left(\mathbf{w}^k - \mathbf{w}^{k-1}\right). \end{aligned} \tag{8}$$

These two quantities provide insights on the upper bound on the absolute error among the objective function and its minimum value at the current iterate [70]. These residuals are employed to design an adaptive strategy for selecting the value for γ, and the convergence of ADMM is assured provided that γ stabilizes after a fixed number of iteration. This strategy is investigated in Sect. 4.5.

4 Numerical Experiments

This section is devoted to assess the performance of the proposed PnPSplit$^+$method. All the experiments have been carried out on a MacBook Pro equipped with M4 processors, in PyTorch environment. The code is available at https://github.com/AleBenfe/PnPSplitPlus.

The images employed for the experiments belong to the Set5 dataset [71] and BSD500 [72]. Each image is scaled to [0, 1]; the Poisson noise has been simulated using a custom function, implemented via `torch` library functions, which allows to select the level ν of the noise affecting the image: the lower the value of ν, the higher the level of the noise. The blurring operation is carried out via FFT.

The network employed as Gaussian denoiser in the update of the variable $\mathbf{w}_2$ in Algorithm 4 is the pretrained deep convolutional network trained in [42], available at https://github.com/basp-group/PnP-MMO-imaging, and depicted in Fig. 1. This network is inspired by the one in [38]: it uses classical convolutional layers and the ReLU layers are replaced by LeakyReLU ones, while the Batch Normalization layers have been removed and a skip connection links the input with the output. This network has been trained by the original authors in two stages:

- A pretraining is carried out on 50 000 images of the ImageNet dataset [73], using the Adam Algorithm, and perturbing each minibatch of images in the following way:

  $$\mathbf{y}_i = \mathbf{x}_i + \sigma_i \varepsilon, \quad \varepsilon \sim \mathcal{N}(0,1),\ \sigma_i \sim \mathcal{U}[0, 0.1].$$

  Therefore, the network is trained to perform *blind denoising*. For this initial training, Jacobian regularization is not encompassed in the loss function (6), that is $\lambda = 0$.
- Once this pretraining is accomplished, a refinement training step inserting Jacobian regularization in (6), considering Remark 2, is done on images of the BSD500 dataset [74], perturbing them with Gaussian noise with a fixed standard deviation equal to 0.01. The resulting network is used in the experiments of this work.

Four different measures are employed to assess the performances of the proposed method: the Mean Square Error (MSE), the relative error (RE) computed as $\|\mathbf{x}^\star - \mathbf{x}^{rec}\|/\|\mathbf{x}^\star\|$, where $\mathbf{x}^{rec}$ is the recovered image, the Peak Signal-to-Noise Ratio and the Structural Similarity Index (SSIM) [75]. These indexes are computed on the last iterate $\mathbf{w}_3^K$: at convergence, the iterates $\mathbf{w}_2^K$, $\mathbf{w}_3^K$ and $\mathbf{x}^K$ should coincide, due to the constraints $\mathbf{M}\mathbf{x} = \mathbf{w}$. The initial iterate $\mathbf{x}^0$ is set equal to $\mathbf{g}$ and all the other variables accordingly: this setting is used among all the numerical experiments. The background b is set to 10^{-3}.

4.1 On the Choice of $\beta\gamma$

Algorithm 4 requires using a denoiser that accounts for the standard deviation $\beta\gamma$ of the Gaussian noise on the current iterate. The parameter β is actually encompassed by the unknown operator A in Proposition 2; hence, the parameter γ plays the role of the standard deviation of the noise. In the numerical experiments presented here, the network employed as a denoiser has been trained on images affected

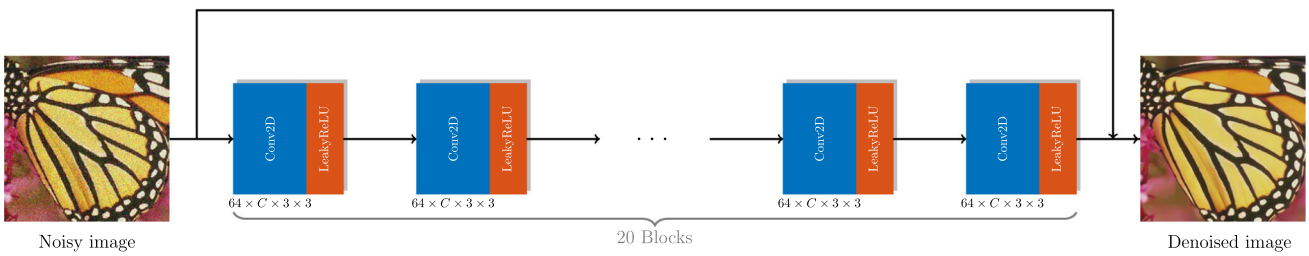


Fig. 1 Sketch of the network used in the experiments. The network consists of 20 blocks of convolutional and LeakyReLU layers, with a skip connection between the input and the output. The number of channels is denoted by C: $C = 3$ for RGB images, $C = 1$ for black&white images

by Gaussian noise with $\sigma = 0.01$: therefore, the choice for the ADMM parameter is set as $\gamma = \sigma = 0.01$.

4.2 Comparison with State-of-the-Art Algorithms

This section is devoted to the comparison with state-of-the-art algorithms. The first run of experiments is carried out for the comparison with the B-PnP algorithm, [55], employing the code provided by the authors in the GitHub repository https://github.com/samuro95/BregmanPnP. Some slight modifications to the original code have been done, in order to run it on the same Apple machine and to have the same Poisson noise generator (`torch.poisson` instead of `numpy.random.poisson`). The comparison has been carried out on high-level Poisson noise ($\nu = 20$), and the images are blurred with a Gaussian PSF with $\sigma = 1$. Both algorithms are tested on two different settings, with the maximum number of iterations set to 2500 and 400. B-PnP uses the PGD algorithm as inner solver.

Table 1 provides the performances indexes on the PSNR, MSE and SSIM. When the maximum number of iteration is reduced to 400 (reflecting the default setting for B-PnP) the methods provide similar results; in particular, B-PnP improves both in terms of visual inspection and of indexes measure. Nonetheless, PnPSplit$^+$achieves higher scores and shows a robust behavior with respect to the number of iterations. Figure 2 presents a visual inspection of the recovered images obtained with 2500 iterations: the ones provided by B-PnP method suffer from the presence of several artifacts, and in the case of the *Butterfly* the image also from some kind of darkening effect.

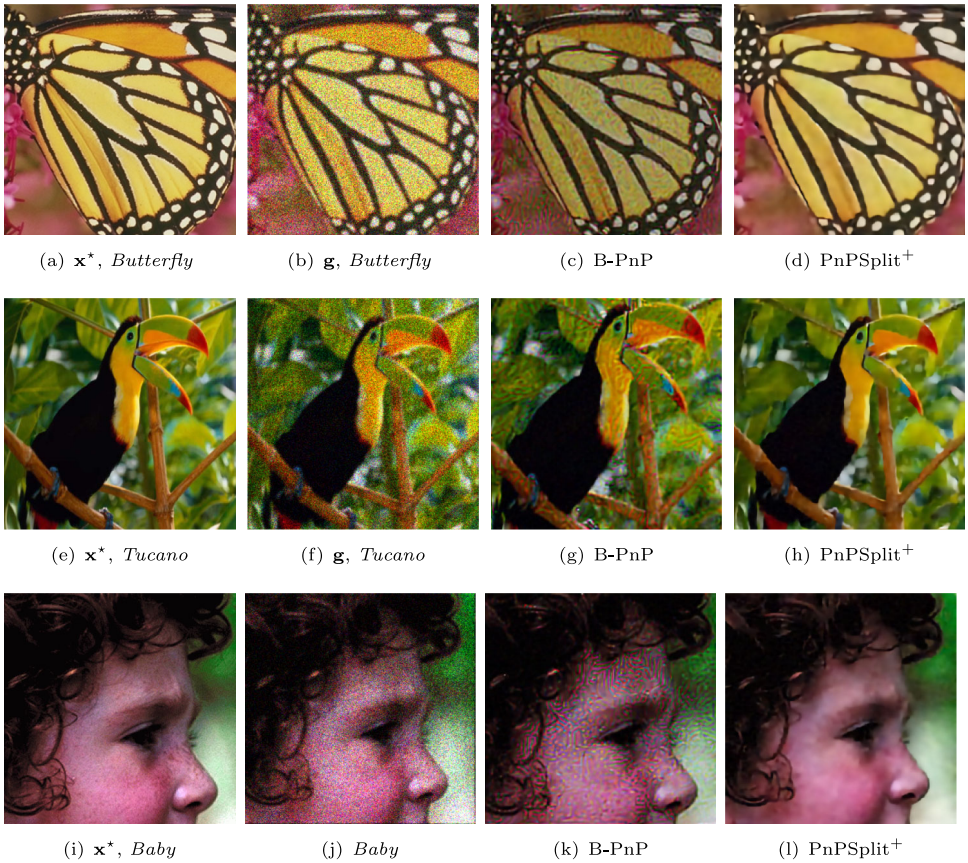
Table 1 Comparison with B-PnP algorithm. Three different images have been considered, namely *Butterfly*, *Tucano* and *Baby*. The proposed algorithm provides reliable performance measures; the B-PnP algorithm achieves better results when the maximum number of iterations is fixed to 400, as per default setting. The proposed method shows robustness with respect to the number of iterations

	2500		400	
	PnPSplit$^+$	B-PnP	PnPSplit$^+$	B-PnP
		PSNR		
Butterfly	25.18	22.14	25.07	23.43
Tucano	29.02	24.86	29.09	28.24
Baby	28.58	21.22	28.21	25.95
		MSE		
Butterfly	0.0030	0.0247	0.0032	0.0045
Tucano	0.0012	0.0043	0.0013	0.0015
Baby	0.0014	0.0090	0.0015	0.0025
		SSIM		
Butterfly	0.8518	0.6464	0.8372	0.7867
Tucano	0.8578	0.7090	0.8539	0.8246
Baby	0.6844	0.4887	0.6789	0.6362

A further comparison with the B-PnP method is carried out on 20 images of the BSD500 dataset [72]; the results are depicted in Fig. 3. These plots show the behavior of the PSNR among the 20 images: for each image 3 different runs have been considered, computing the average value and standard deviation for the PSNR, in order to avoid that a particular noise realization affects the performance evaluation. The shaded curved area represents the average PSNR $\pm$ the standard deviation. Two experiments were performed. In Fig. 3a, the maximum number of iterations was set to 1000 for both methods, whereas in Fig. 3a, it was set to 400. PnPSplit$^+$is again more robust to the choice of maximum number of iterations, and B-PnP performs better when it is run for a low number of iterations, as presented in the original paper [55]. The computational time of PnPSplit$^+$method is lower than the one of B-PnP: the latter takes on average 7 seconds to run for 400 iterations, while B-PnP needs around 24 seconds. The difference increases for a larger number of iterations.

The third run of experiments is done for comparing the PnPSplit$^+$with two other approaches: QAB-PnP [56] and P^{4}IP [57]. The test images employed in these experiments are modifications of the original ones, due to the memory constraints posed by the available MATLAB code for QAB-PnP: the images are halved in both dimensions and transformed in grayscale images. The PSF inducing the blur is still a Gaussian one with $\sigma = 1$ and the noise level is set to 20. The denoiser used in Algorithm 4 is again the one from [42] with the appropriate number of input channels. Algorithm QAB-PnP runs on MATLAB with no parallel implementation, the code is available at https://github.com/SayantanDutta95/QAB-PnP-ADMM-Deconvolution, while the Python code for P^{4}IP can be downloaded at https://github.com/sanghviyashiitb/poisson-plug-and-play/tree/main.

Table 2 presents the numerical assessment of the performance of the three algorithms. Figure 4 shows the recorded data $\mathbf{g}$ for the three images, together with the recovered images achieved by the three different algorithms. The effect of the PSF is significant, given the images' dimension, and the noise level is rather high. The restored images achieved by QAB-PnP present several artifacts, while the ones provided by PnPSplt+ suffer from the loss of details, mainly in *Baby* cases. P^{4}IP failed to recover reliable images: the images shown in Fig. 4 related to the result of P^{4}IP are rescaled in order to make them visible, since the reached maximum value is around 0.04 in all three cases.

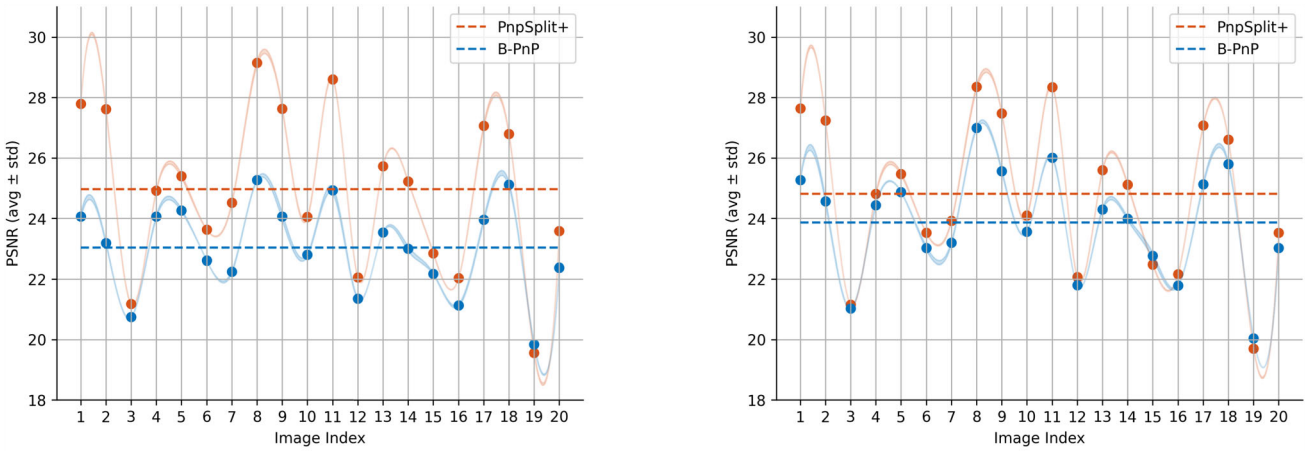


(a) $\mathbf{x}^\star$, *Butterfly* (b) $\mathbf{g}$, *Butterfly* (c) B-PnP (d) PnPSplit$^+$

(e) $\mathbf{x}^\star$, *Tucano* (f) $\mathbf{g}$, *Tucano* (g) B-PnP (h) PnPSplit$^+$

(i) $\mathbf{x}^\star$, *Baby* (j) *Baby* (k) B-PnP (l) PnPSplit$^+$

Fig. 2 Visual inspection of the recovered images provided by PnPSplit$^+$and B-PnP algorithms. First column: ground truth images. Second column: simulated recorded data, perturbed with a Gaussian PSF and Poisson noise at level 20. Third column: B-PnP reconstruction. Fourth column: PnPSplit$^+$reconstruction. Both algorithms have run for 2500 iterations. The B-PnP reconstructions suffer from the presence of some artifacts, while PnPSplit$^+$ones presents more smooth results

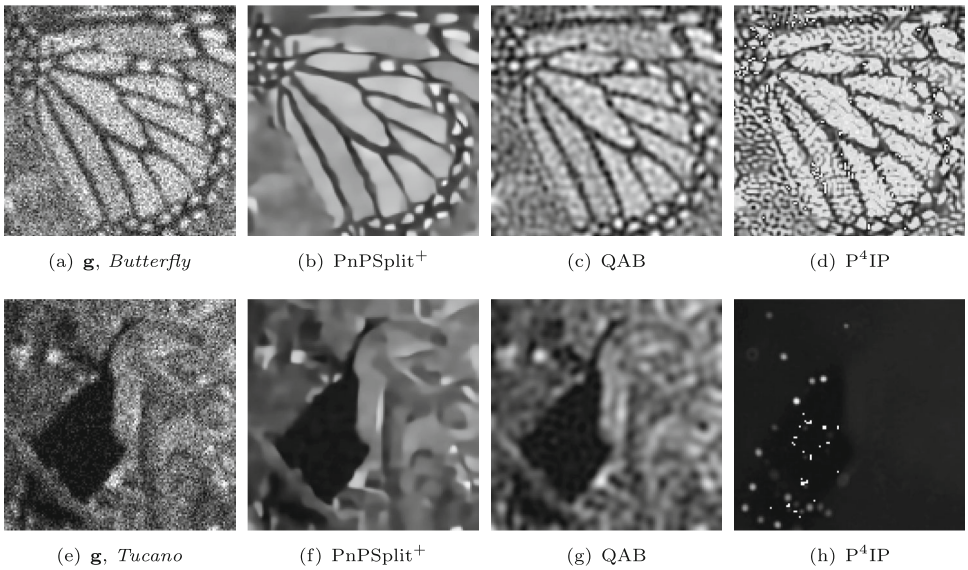


(a) Max iters: 1000 (b) Max Iters: 400

Fig. 3 PSNR assessment on 20 images of the BSDS500 dataset. The panels show the results for PnPSplit$^+$and B-PnP when the maximum number of iteration is set to 1000 (left) and to 400 (right). The dots represent the average PSNR of the recovered images for each method. PnPSplit$^+$reveals to be quite robust with respect to the maximum number of iterations, and even when B-PnP runs with the optimal number of iterations, PnPSplit$^+$is competitive. The curved behavior is due to the cubic spline used for plotting the results

Table 2 Performances of PnPSplit$^+$, QAB-PnP and P^{4}IP algorithms on grayscale images corrupted by a Gaussian PSF with $\sigma = 1$ and $\nu = 20$. PnPSplit$^+$provides better results than QAB-PnP. P^{4}IP instead does not reach reliable results and suffers particularly from the presence of noise

	PnPSplit$^+$			QAB-PnP			P^{4}IP		
	Butterfly	*Tucano*	*Baby*	*Butterfly*	*Tucano*	*Baby*	*Butterfly*	*Tucano*	*Baby*
MSE	0.0057	0.0026	0.0012	0.1245	0.0621	0.0413	0.2213	0.1255	0.0862
RE	0.1526	0.1648	0.0937	0.2202	0.1902	0.1086	0.9512	0.9478	0.9478
PSNR	22.444	25.838	29.110	18.094	24.135	27.689	6.5497	9.012	10.640
SSIM	0.8104	0.7487	0.8006	0.5312	0.6551	0.7441	0.0262	0.1193	0.0641



(a) $\mathbf{g}$, *Butterfly* (b) PnPSplit$^+$ (c) QAB (d) P^{4}IP

(e) $\mathbf{g}$, *Tucano* (f) PnPSplit$^+$ (g) QAB (h) P^{4}IP

Fig. 4 Comparison on the reconstruction achieved by PnPSplit$^+$, QAB-PNP and P^{4}IP, respectively, on the second, third and fourth column. The first column shows the corrupted data $\mathbf{g}$. The results of P^{4}IP are shown in a different scale: while PnPSplit$^+$and QAB provide reconstructions in [0,1], P^{4}IP failed to recover images with values higher than 0.04 in all cases

4.3 Severely Corrupted Images

The following set of experiment is devoted to assess the performance of the PnPSplit$^+$Algorithm in presence of high noise level or severe blur induced by the PSF.

Table 3 (upper part) presents the numerical performance of PnPSplit$^+$when the Poisson noise level ν is increased to 15 and 10. As one expects, the higher the noise level the worst the performances, but nonetheless the achieved results present rather high scores: in particular, the PSNR of the recovered images reaches satisfying levels. Table 3 (lower part) shows the four scores achieved when large Gaussian PSF ($\sigma = 2$ and $\sigma = 2.5$) are used to blur the images, with $\nu = 20$. The quality of the restoration is reliable, although in this case the information loss induced by the blurring is too high to retrieve pleasant images to the human eyes.

Figure 5 presents the recovered images when the noise level is set to 5 and when the PSF inducing the blurring is large ($\sigma = 2.5$) for *Butterfly*, *Tucano* and *Baby* images on the first, second and third row, respectively. As one expects, the recovered images present several artifacts, mainly when recovering in presence of high noise, but even in these extreme cases Algorithm 4 manages to recover most of the information.

Table 3 Results achieved by PnPSplit$^+$when the Gaussian PSF induces a larger blur and for high-level noise. The loss of information is relatively high, but the index measures are still reliable

	Butterfly	*Tucano*	*Baby*	*Butterfly*	*Tucano*	*Baby*
		$\nu = 15$			$\nu = 10$	
MSE	0.0036	0.0016	0.0018	0.0088	0.0096	0.0049
RE	0.1135	0.1135	0.1110	0.1773	0.2787	0.1818
PSNR	24.425	27.987	27.380	20.555	20.185	23.095
SSIM	0.8369	0.8342	0.6717	0.7533	0.7020	0.6025
		$\sigma = 2$			$\sigma = 2.5$	
MSE	0.0062	0.0021	0.0018	0.0078	0.0028	0.0020
RE	0.1493	0.1316	0.1096	0.1671	0.1494	0.1150
PSNR	22.047	26.706	27.488	21.071	25.601	27.077
SSIM	0.7588	0.7907	0.6332	0.7172	0.7562	0.6160

4.4 Other Blurring Operators

The proposed PnPSplit$^+$approach is tested on other blur operators, namely

- Motion blur, with length 11 and slope 35°,
- Out-of-Focus blur, with radius 5,
- Mean Filter blur, with dimension equal to 3.



(a) $\mathbf{g}$, $\nu = 10$ (b) Recovered image (c) $\mathbf{g}$, $\sigma = 2.5$ (d) Recovered image

(e) $\mathbf{g}$, $\nu = 10$ (f) Recovered image (g) $\mathbf{g}$ $\sigma = 2.5$ (h) Recovered image

(i) $\mathbf{g}$, $\nu = 10$ (j) Recovered image (k) $\mathbf{g}$, $\sigma = 2.5$ (l) Recovered image

Fig. 5 Image results when the perturbation on the recorded data is particularly strong, in terms of noise level or blurring. First row: reconstructions obtained for a PSF with $\sigma = 1$ and noise level set to 5. Second row: reconstructions obtained for a PSF with $\sigma = 2.5$ and noise level set to 20



(a) Motion Blur (b) Out-of-Focus Blur (c) Mean Filter Blur

(d) PSNR 23.052 (e) PSNR 25.283 (f) PSNR 28.626

Fig. 6 Result of PnPSplit$^+$ for Motion, Out-of-Focus and Mean filter blurs. Top row: corrupted images; bottom row: recovered images with relative PSNR

The corrupted data $\mathbf{g}$ are computed using the linear model (1) employing one of the above operators for $\mathbf{H}$, and the noise level is set again to 20. The parameter γ is set to 0.01, and the method is run for 1000 iterations. The corrupted data and the relative restorations are depicted in Fig. 6, together with the numerical performance metrics. PnPSplit$^+$provides reliable results under different types of blurring operators.

4.5 Adaptive Strategy for γ

As observed in Remark 4, ADMM performance could be particularly dependent on the choice of γ. The literature presents several adaptive strategies to overcome this issue: this works employs the one depicted in [68–70], and Algorithm 4 can be

Table 4 Evaluation of fixed versus adaptive strategy. The column γ^0 denotes the value for γ selected as initial one for the adaptive strategy (Adapt column) and the constant used in the vanilla PnPSplit$^+$. The index measures of Mean Square Error, Relative Error, Peak Signal-to-Noise Ratio and Similarity Structure Index Measure are employed for the comparison. The adaptive strategy is particularly effective for high values of γ

	Butterfly							
	$\gamma^0 : 1$	Adapt	$\gamma^0 : 10$	Adapt	$\gamma^0 : 10^2$	Adapt	$\gamma^0 : 10^3$	Adapt
MSE	0.1132	0.1129	0.1272	0.0944	0.1293	0.0249	0.1295	0.0035
RE	0.6360	0.6349	0.6739	0.5806	0.6795	0.2982	0.6800	0.1121
PSNR	9.461	9.475	8.957	10.252	8.886	16.039	8.8879	24.538
SSIM	0.1686	0.1690	0.1546	0.1902	0.1538	0.3941	0.1528	0.8328
	$\gamma^0 : 10^{-4}$	Adapt	$\gamma^0 : 10^{-3}$	Adapt	$\gamma^0 : 10^{-2}$	Adapt	$\gamma^0 : 10^{-1}$	Adapt
MSE	0.0150	0.0127	0.0086	0.0047	0.0031	0.0072	0.0755	0.0871
RE	0.2312	0.2132	0.1754	0.1295	0.1046	0.1604	0.5193	0.5577
PSNR	18.248	18.953	20.648	23.284	25.137	21.425	11.220	10.600
SSIM	0.6302	0.6609	0.7267	0.8024	0.8526	0.6629	0.2263	0.2026
	Tucano							
	$\gamma^0 : 10^{-4}$	Adapt	$\gamma^0 : 10^{-3}$	Adapt	$\gamma^0 : 10^{-2}$	Adapt	$\gamma^0 : 10^{-1}$	Adapt
MSE	0.0098	0.0078	0.0047	0.0020	0.0013	0.0048	0.0577	0.0693
RE	0.2812	0.2507	0.1947	0.1283	0.1017	0.1971	0.6842	0.7497
PSNR	20.108	21.104	23.302	26.921	28.946	23.195	12.385	11.591
SSIM	0.5378	0.6021	0.7043	0.8147	0.8582	0.6531	0.2509	0.1894
	$\gamma^0 : 1$	Adapt	$\gamma^0 : 10$	Adapt	$\gamma^0 : 10^2$	Adapt	$\gamma^0 : 10^3$	Adapt
MSE	0.0859	0.0884	0.0986	0.0726	0.1001	0.0160	0.1003	0.0014
RE	0.8344	0.8465	0.8939	0.7669	0.9008	0.3602	0.9016	0.1079
PSNR	10.660	10.536	10.063	11.393	9.996	17.957	9.988	28.426
SSIM	0.1424	0.1375	0.1272	0.1618	0.1411	0.4119	0.1244	0.8478

modified inserting the following γ scheduler after the update of the Lagrangian parameters.

$$\gamma^{k+1} = \begin{cases} \dfrac{\alpha}{\gamma^k} & \text{if } \|\mathbf{p}^k\| > \mu\|\mathbf{d}^k\|,\ k \le k_{max} \\ \alpha\gamma^k & \text{if } \|\mathbf{d}^k\| > \mu\|\mathbf{p}^k\|,\ k \le k_{max} \\ \gamma^k & \text{otherwise} \end{cases} \tag{9}$$

where α and μ are positive values greater than 1. A first glance, it seems that the number of parameters to set rises from one to four: actually, α and μ can be set really close to 1 and the only parameters to set remain γ^0 and k_{max}. A first experiment is carried out for testing the relevance of the adaptive strategy for γ, and how the initial parameter influences Algorithm 4 results.

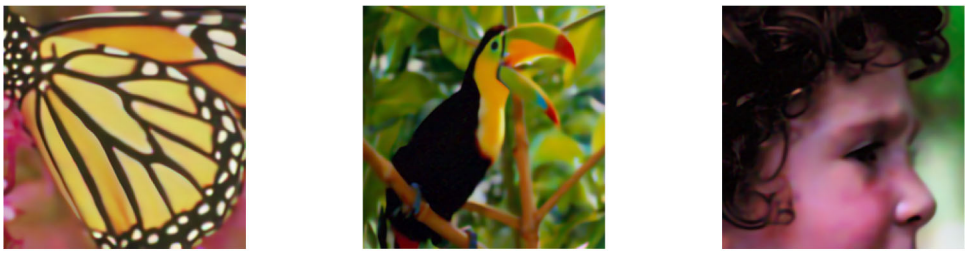
Two images, namely *Butterfly* and *Tucano*, are employed for this test: each one is blurred with a Gaussian PSF with standard deviation $\sigma = 1$ and corrupted with Poisson noise at level $\nu = 20$. For each choice γ^0, Algorithm 4 is run also with $\gamma^k = \gamma^0$ for any k. The maximum number K of iteration is set to 2500, $\alpha = \mu = 1.001$ and $k_{max} = 1250$.

Table 4 shows the comparison result. Considering a constant value for γ, the best choice is $\gamma = 0.01$, which is exactly the parameter used for the noisy images employed for the training of D_{ϑ} used in Algorithm 4. One should note that the adaptive strategy when $\gamma^0 = 0.01$ fails, while in most all the other cases such strategy improves the quality of the restored images. These tests confirm that choosing the constant value $\gamma = 0.01$ provides reliable performances under the usage of the denoiser presented in [42]. If one employs a denoiser whose training is carried out on noisy images with different noise levels, then using the adaptive strategy could overcome possible issues: in Sect. 4.6 this approach shall reveal to provide remarkable results.

4.6 Performance Without Convergence Guarantees

The last runs of experiments consist of the implementation of Algorithm 4 when the denoising network is not firmly non-expansive, *i.e.*, not abiding to the hypothesis that guarantees the convergence of the method. The network trained in [42] is substituted by the classical deep convolutional network presented in [38]. Such network has been trained by minimizing the well-known MSE loss function, therefore without imposing any constraint that forces the non firmly expansiveness.

The results are collected in Fig. 7 and the numerical performance is summed up in Table 5, for the case in which the blur is induced by a Gaussian PSF with $\sigma = 1$ and the noise level is set to 20. Setting a fixed value for γ does not yield to reliable results, likely due to the absence of convergence property. Therefore, these tests are run employing the adaptive strategy depicted in Sect. 4.5, setting the initial value



(a) *Butterfly* (b) *Tucano* (c) *Baby*

Fig. 7 Recovered images when the convergence guarantees are not met. The quality of these reconstructions is similar to the quality of the images obtained employing a net satisfying the convergence guarantees, both in terms of visual inspection and performance measures

Table 5 Numerical assessment of the achieved results when a net not satisfying the requirements of non-firmly expansion is not met. The indexes values are slightly lower than the ones obtained in Sect. 4.2: this could be due to the denoising network

	Butterfly	*Tucano*	*Baby*
MSE	0.0050	0.0015	0.0015
RE	0.1337	0.1113	0.1017
PSNR	23.003	28.163	28.137
SSIM	0.7894	0.8267	0.6528

of γ to 10, the maximum number of iterations to 1000 and $k_{max} = 1000$ in (9).

The numerical experience shows that even employing a network that, at a first glance, does not assure the convergence of the method allows to achieve reliable results, both in terms of visual inspection and of measurement indexes. The latter ones do not achieve values close to the ones in Table 1: this could be due to the quality of denoising ability of the network.

5 Conclusion

This work presented a novel approach, named PnPSplit$^+$, for solving image restoration problems in presence of Poisson noise. The original idea of [59] is coupled with PnP strategy of substituting the proximal step on the regularization function with an off-the-shelf denoiser. In particular, for ensuring the convergence of the method a firmly non-expansive denoiser has been employed in the PnPSplit$^+$scheme. The main contribution of this approach is to avoid the usage of an inner solver for the deblurring step, allowing the computation of solution to the inner problem via an explicit formula. This strategy showed remarkable performances, both in terms of quality measurements and computational time, in comparison to state-of-the-art algorithms, and even in presence of high noise levels and when the blurring effect of the PSF is significant.

The results are really promising, but nonetheless there are still several aspects to explore and improve. On the one hand, the adaptive strategy should be further investigated for both firmly non-expansive denoisers and those that do not satisfy convergence guarantees. In the latter case, the adaptive strategy may play a role in the convergence behavior of the method. Another aspect to be considered in future research is the employment of firmly non-expansive blind denoising networks, that is networks trained for the noise removal with different standard deviations. A further research direction could consider the development of a suitable stopping criterion, aiming to avoid early stopping via the setting the maximum number of iteration. Finally, a further generalization of the proposed approach can be done in the direction of Proximal Gradient Descent Ascent methods.

Acknowledgements This research has been partially performed in the framework of the MIUR-PRIN Grant 20225STXSB "Sustainable Tomographic Imaging with Learning and Regularization," GNCS Project CUP E53C24001950001 "Metodi avanzati di ottimizzazione stocastica per la risoluzione di problemi inversi di imaging," and CARIPLO project "Project Data Science Approach for Carbon Farming Scenarios (DaSACaF)"CAR_RIC25ABENF_01.

Author Contributions Alessandro Benfenati helped in conceptualization, methodology, formal analysis and investigation, writing—original draft preparation, writing—review and editing, funding acquisition, resources, supervision.

Funding Open access funding provided by Università degli Studi di Milano within the CRUI-CARE Agreement.

Data Availability Data are provided within the manuscript or supplementary information files

Declarations

Competing interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Loli Piccolomini, E., Prato, M., Scipione, M., Sebastiani, A.: CTprintNet: an accurate and stable deep unfolding approach for few-view CT reconstruction. Algorithms (2023). https://doi.org/10.3390/a16060270
2. Hansen, P.C., Jorgensen, J., Lionheart, W.R.B.: Computed Tomography: Algorithms, Insight, and Just Enough Theory. Society for Industrial and Applied Mathematics, Philadelphia (2021). https://doi.org/10.1137/1.9781611976670
3. Beltramo-Martin, O., Fétick, R., Neichel, B., Fusco, T.: Joint estimation of atmospheric and instrumental defects using a parsimonious point spread function model-on-sky validation using state of the art worldwide adaptive-optics assisted instruments. Astron. Astrophys. **643**, 58 (2020). https://doi.org/10.1051/0004-6361/202038679
4. Dabbech, A., Terris, M., Jackson, A., Ramatsoku, M., Smirnov, O.M., Wiaux, Y.: First AI for deep super-resolution wide-field imaging in radio astronomy: unveiling structure in ESO 137–006. Astrophys. J. Lett. **939**(1), 4 (2022). https://doi.org/10.3847/2041-8213/ac98af
5. Benfenati, A., La Camera, A., Carbillet, M.: Deconvolution of post-adaptive optics images of faint circumstellar environments by means of the inexact Bregman procedure. Astron. Astrophys. **586**, 16 (2016). https://doi.org/10.1051/0004-6361/201526960
6. Calisesi, G., Ghezzi, A., Ancora, D., D'Andrea, C., Valentini, G., Farina, A., Bassi, A.: Compressed sensing in fluorescence microscopy. Prog. Biophys. Mol. Biol. **168**, 66–80 (2022). https://doi.org/10.1016/j.pbiomolbio.2021.06.004
7. Zunino, A., Castello, M., Vicidomini, G.: Reconstructing the image scanning microscopy dataset: an inverse problem. Inverse Prob. **39**(6), 064004 (2023). https://doi.org/10.1088/1361-6420/accdc5
8. Cascarano, P., Comes, M.C., Sebastiani, A., Mencattini, A., Loli Piccolomini, E., Martinelli, E.: DeepCEL0 for 2D single-molecule localization in fluorescence microscopy. Bioinformatics **38**(5), 1411–1419 (2021). https://doi.org/10.1093/bioinformatics/btab808
9. Bertero, M., Boccacci, P., De Mol, C.: Introduction to Inverse Problems in Imaging. CRC Press, Boca Raton (2021)
10. Zhong, Q., Yin, K., Duan, Y.: Image reconstruction by minimizing curvatures on image surface. J. Math. Imaging Vis. **63**(1), 30–55 (2021). https://doi.org/10.1007/s10851-020-00992-3
11. Cheng, L., Xing, Y., Li, Y., Guo, Z.: A diffusion equation for improving the robustness of deep learning speckle removal model. J. Math. Imaging Vis. **66**(5), 801–821 (2024). https://doi.org/10.1007/s10851-024-01199-6
12. Bertero, M., Boccacci, P., Ruggiero, V.: Inverse Imaging with Poisson Data. 2053–2563. IOP Publishing, Bristol, England, UK (2018). https://doi.org/10.1088/2053-2563/aae109
13. Lanza, A., Morigi, S., Sgallari, F., Wen, Y.-W.: Image restoration with Poisson Gaussian mixed noise. Comput. Methods Biomech. Biomed. Eng. Imaging Visual. **2**(1), 12–24 (2014). https://doi.org/10.1080/21681163.2013.811039
14. Vardi, Y., Shepp, L.A., Kaufman, L.: A statistical model for positron emission tomography. J. Am. Stat. Assoc. **80**(389), 8–20 (1985). https://doi.org/10.1080/01621459.1985.10477119
15. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. PAMI- **6**(6), 721–741 (1984). https://doi.org/10.1109/TPAMI.1984.4767596
16. Tikhonov, A.N.: Solution of incorrectly formulated problems and the regularization method. Sov Dok **4**, 1035–1038 (1963)
17. Golub, G.H., Von Matt, U.: Tikhonov regularization for large scale problems. In: Scientific Computing: Proceedings of the Workshop, 10–12 March 1997, Hong Kong, p. 3. Springer, Berlin (1998)https://doi.org/10.1023/A:1026083619097
18. Arashi, M., Roozbeh, M., Hamzah, N.A., Gasparini, M.: Ridge regression and its applications in genetic studies. PLoS ONE **16**(4), 1–17 (2021). https://doi.org/10.1371/journal.pone.0245376
19. Gapyak, V., Rentschler, C.E., März, T., Weinmann, A.: An ℓ_1-plug-and-play approach for MPI using a zero shot denoiser with evaluation on the 3D open MPI dataset. Phys. Med. Biol. **70**(2), 025028 (2025). https://doi.org/10.1088/1361-6560/ada5a1
20. Benfenati, A., Causin, P., Lupieri, M., Naldi, G.: Regularization techniques for inverse problem in DOT applications. J. Phys. Conf. Ser. **1476**, 012007 (2020). https://doi.org/10.1088/1742-6596/1476/1/012007
21. Caselles, V., Chambolle, A., Novaga, M.: Total variation in imaging. In: Handbook of Mathematical Methods in Imaging, pp. 1455–1499. Springer, New York, NY (2015). https://doi.org/10.1007/978-0-387-92920-0_23
22. Morotti, E., Evangelista, D., Sebastiani, A., Piccolomini, E.L.: Space-variant total variation boosted by learning techniques in few-view tomographic imaging. (2024) arXiv:2404.16900
23. Landi, G., Viola, M., Zama, F.: A scaled gradient projection method for the realization of the balancing principle in TGV-based image restoration. Comput. Optim. Appl. **91**(2), 759–785 (2025). https://doi.org/10.1007/s10589-025-00659-y
24. Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. Acta Numer. **25**, 161–319 (2016). https://doi.org/10.1017/S096249291600009X
25. Crisci, S., Porta, F., Ruggiero, V., Zanni, L.: Hybrid limited memory gradient projection methods for box-constrained optimization problems. Comput. Optim. Appl. **84**(1), 151–189 (2023). https://doi.org/10.1007/s10589-022-00409-4
26. Zhang, J., Hu, Y., Nagy, J.G.: A scaled gradient method for digital tomographic image reconstruction. Inverse Prob. Imaging **12**(1), 239–259 (2018). https://doi.org/10.3934/ipi.2018010
27. Savanier, M., Chouzenoux, E., Pesquet, J.-C., Riddell, C.: Unmatched preconditioning of the proximal gradient algorithm. IEEE Signal Process. Lett. **29**, 1122–1126 (2022). https://doi.org/10.1109/LSP.2022.3169088
28. Li, H., Zhang, W., Huang, S., Xiao, F.: Poisson noise image restoration based on Bregman proximal gradient. In: 2023 6th International Conference on Computer Network, Electronic and Automation (ICCNEA), pp. 239–243 (2023). IEEE https://doi.org/10.1109/ICCNEA60107.2023.00058
29. Benfenati, A., Ruggiero, V.: Inexact Bregman iteration with an application to Poisson data reconstruction. Inverse Prob. **29**(6), 065016 (2013). https://doi.org/10.1088/0266-5611/29/6/065016
30. Rossignol, C., Sureau, F., Chouzenoux, É., Comtat, C., Pesquet, J.-C.: A Bregman majorization-minimization framework for PET image reconstruction. In: 2022 IEEE International Conference on Image Processing (ICIP), pp. 1736–1740 (2022). IEEE https://doi.org/10.1109/ICIP46576.2022.9897619
31. Chouzenoux, E., Corbineau, M.-C., Pesquet, J.-C., Scrivanti, G.: A variational approach for joint image recovery and feature extraction based on spatially varying generalised Gaussian models. J. Math. Imaging Vis. **66**(4), 550–571 (2024). https://doi.org/10.1007/s10851-024-01184-z
32. Figueiredo, M.A., Bioucas-Dias, J.M.: Restoration of Poissonian images using alternating direction optimization. IEEE Trans. Image Process. **19**(12), 3133–3145 (2010). https://doi.org/10.1109/TIP.2010.2053941
33. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al., Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends® Mach. Learn. **3**(1), 1–122 (2011) https://doi.org/10.1561/2200000016

34. Serafino, D., Landi, G., Viola, M.: Directional TGV-based image restoration under Poisson noise. J. Imaging **7**(6), 99 (2021). https://doi.org/10.3390/jimaging7060099
35. Venkatakrishnan, S.V., Bouman, C.A., Wohlberg, B.: Plug-and-play priors for model based reconstruction. In: 2013 IEEE Global Conference on Signal and Information Processing, pp. 945–948 (2013). https://doi.org/10.1109/GlobalSIP.2013.6737048
36. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (2007). https://doi.org/10.1109/TIP.2007.901238
37. Buades, A., Coll, B., Morel, J.-M.: Non-local means denoising. Image Process. Line **1**, 208–212 (2011). https://doi.org/10.5201/ipol.2011.bcm_nlm
38. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans. Image Process. **26**(7), 3142–3155 (2017). https://doi.org/10.1109/TIP.2017.2662206
39. Ryu, E., Liu, J., Wang, S., Chen, X., Wang, Z., Yin, W.: Plug-and-play methods provably converge with properly trained denoisers. In: International Conference on Machine Learning, pp. 5546–5557 (2019). PMLR
40. Terris, M., Repetti, A., Pesquet, J.-C., Wiaux, Y.: Building firmly nonexpansive convolutional neural networks. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8658–8662 (2020). https://doi.org/10.1109/ICASSP40776.2020.9054731
41. Combettes, P.L., Pesquet, J.-C.: Deep neural network structures solving variational inequalities. Set-Valued Var. Anal. **28**(3), 491–518 (2020). https://doi.org/10.1007/s11228-019-00526-z
42. Pesquet, J.-C., Repetti, A., Terris, M., Wiaux, Y.: Learning maximally monotone operators for image recovery. SIAM J. Imag. Sci. **14**(3), 1206–1237 (2021). https://doi.org/10.1137/20M1387961
43. Ducotterd, S., Goujon, A., Bohra, P., Perdios, D., Neumayer, S., Unser, M.: Improving Lipschitz-constrained neural networks by learning activation functions. J. Mach. Learn. Res. **25**(65), 1–30 (2024)
44. Hertrich, J., Neumayer, S., Steidl, G.: Convolutional proximal neural networks and plug-and-play algorithms. Linear Algebra Appl. **631**, 203–234 (2021). https://doi.org/10.1016/j.laa.2021.09.004
45. Yoshida, Y., Miyato, T.: Spectral norm regularization for improving the generalizability of deep learning (2017). arXiv:1705.10941
46. Benfenati, A., Cascarano, P.: Constrained plug-and-play priors for image restoration. J. Imaging (2024). https://doi.org/10.3390/jimaging10020050
47. Chan, S.H., Wang, X., Elgendy, O.A.: Plug-and-play ADMM for image restoration: fixed-point convergence and applications. IEEE Trans. Comput. Imaging **3**(1), 84–98 (2017). https://doi.org/10.1109/TCI.2016.2629286
48. Sun, Y., Wu, Z., Xu, X., Wohlberg, B., Kamilov, U.S.: Scalable plug-and-play ADMM with convergence guarantees. IEEE Trans. Comput. Imaging **7**, 849–863 (2021). https://doi.org/10.1109/TCI.2021.3094062
49. Nair, P., Gavaskar, R.G., Chaudhury, K.N.: Fixed-point and objective convergence of plug-and-play algorithms. IEEE Trans. Comput. Imaging **7**, 337–348 (2021). https://doi.org/10.1109/TCI.2021.3066053
50. Romano, Y., Elad, M., Milanfar, P.: The little engine that could: regularization by denoising (RED). SIAM J. Imag. Sci. **10**(4), 1804–1844 (2017). https://doi.org/10.1137/16M1102884
51. Cohen, R., Elad, M., Milanfar, P.: Regularization by denoising via fixed-point projection (RED-PRO). SIAM J. Imag. Sci. **14**(3), 1374–1406 (2021). https://doi.org/10.1137/20M1337168
52. Cascarano, P., Benfenati, A., Kamilov, U.S., Xu, X.: Constrained regularization by denoising with automatic parameter selection. IEEE Signal Process. Lett. **31**, 556–560 (2024). https://doi.org/10.1109/LSP.2024.3359569
53. Daniele, C., Villa, S., Vaiter, S., Calatroni, L.: Deep equilibrium models for Poisson imaging inverse problems via mirror descent (2025). arXiv:2507.11461
54. Hurault, S., Leclaire, A., Papadakis, N.: Gradient step denoiser for convergent plug-and-play. In: International Conference on Learning Representations (2022). https://openreview.net/forum?id=fPhKeld3Okz
55. Hurault, S., Kamilov, U., Leclaire, A., Papadakis, N.: Convergent Bregman plug-and-play image restoration for Poisson inverse problems. Adv. Neural. Inf. Process. Syst. **36**, 27251–27280 (2023)
56. Dutta, S., Basarab, A., Georgeot, B., Kouam , D.: Poisson image deconvolution by a plug-and-play quantum denoising scheme. In: 2021 29th European Signal Processing Conference (EUSIPCO), pp. 646–650 (2021). https://doi.org/10.23919/EUSIPCO54536.2021.9616253
57. Rond, A., Giryes, R., Elad, M.: Poisson inverse problems by the plug-and-play scheme. J. Vis. Commun. Image Represent. **41**, 96–108 (2016). https://doi.org/10.1016/j.jvcir.2016.09.009
58. Klatzer, T., Melidonis, S., Pereyra, M., Zygalakis, K.C.: Efficient Bayesian computation using plug-and-play priors for Poisson inverse problems (2025). arXiv:2503.16222
59. Setzer, S., Steidl, G., Teuber, T.: Deblurring Poissonian images by split Bregman techniques. J. Vis. Commun. Image Represent. **21**(3), 193–199 (2010). https://doi.org/10.1016/j.jvcir.2009.10.006
60. Cascarano, P., Piccolomini, E.L., Morotti, E., Sebastiani, A.: Plug-and-play gradient-based denoisers applied to CT image enhancement. Appl. Math. Comput. **422**, 126967 (2022). https://doi.org/10.1016/j.amc.2022.126967
61. Pendu, M.L., Guillemot, C.: Preconditioned plug-and-play ADMM with locally adjustable denoiser for image restoration. SIAM J. Imag. Sci. **16**(1), 393–422 (2023). https://doi.org/10.1137/22M1504809
62. Gavaskar, R.G., Athalye, C.D., Chaudhury, K.N.: On plug-and-play regularization using linear denoisers. IEEE Trans. Image Process. **30**, 4802–4813 (2021). https://doi.org/10.1109/TIP.2021.3075092
63. Kamilov, U.S., Bouman, C.A., Buzzard, G.T., Wohlberg, B.: Plug-and-play methods for integrating physical and learned models in computational imaging: theory, algorithms, and applications. IEEE Signal Process. Mag. **40**(1), 85–97 (2023). https://doi.org/10.1109/MSP.2022.3199595
64. Bolte, J., Pauwels, E.: Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning. Math. Program. **188**, 19–51 (2021). https://doi.org/10.1007/s10107-020-01501-5
65. Bolte, J., Le, T., Pauwels, E., Silveti-Falls, A.: Nonsmooth implicit differentiation for machine learning and optimization. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. NIPS '21. Curran Associates Inc., Red Hook, NY, USA (2021)
66. Bolte, J., Pauwels, E., Vaiter, S.: Automatic differentiation of nonsmooth iterative algorithms. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22. Curran Associates Inc., Red Hook, NY, USA (2022)
67. Bolte, J., Pauwels, E., Silveti-Falls, A.: Differentiating nonsmooth solutions to parametric monotone inclusion problems. SIAM J. Optim. **34**(1), 71–97 (2024). https://doi.org/10.1137/22M1541630
68. He, B.S., Yang, H., Wang, S.L.: Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. J. Optim. Theory Appl. **106**(2), 337–356 (2000). https://doi.org/10.1023/A:1004603514434
69. Wang, S.L., Liao, L.Z.: Decomposition method with a variable parameter for a class of monotone variational inequality problems.

J. Optim. Theory Appl. **109**(2), 415–429 (2001). https://doi.org/10.1023/A:1017522623963
70. Zanni, L., Benfenati, A., Bertero, M., Ruggiero, V.: Numerical methods for parameter estimation in Poisson data inversion. J. Math. Imaging Vis. **52**(3), 397–413 (2015). https://doi.org/10.1007/s10851-014-0553-9
71. Bevilacqua, M., Roumy, A., Guillemot, C.M., Alberi-Morel, M.-L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: British Machine Vision Conference (2012). https://doi.org/10.5244/C.26.135
72. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 898–916 (2011). https://doi.org/10.1109/TPAMI.2010.161
73. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
74. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, vol. 2, pp. 416–4232 (2001). https://doi.org/10.1109/ICCV.2001.937655
75. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.