PDE-constrained optimisation problem for image denoising and weight parameter identification

Alexandre Boistard

Mathematics and Data Science, Data and Information Science, CentraleSupélec

February 28, 2025

Abstract

This project explores the application of optimization techniques in computer vision by analyzing and implementing key concepts from the paper *A Weighted Parameter Identification PDE-Constrained Optimization for Inverse Image Denoising Problem*. The paper introduces a novel approach to image denoising by formulating the problem as a PDE-constrained optimization task. Unlike traditional methods that rely on preselected fixed parameters, this approach estimates both the clean image and the spatially varying weighting parameter simultaneously using an optimization framework. This method ensures enhanced preservation of edges and adaptive smoothing in homogeneous regions. The project includes a synthesis report detailing the methodology and contributions of the paper, followed by an implementation of its core experiences in a Jupyter Notebook.

Keywords: Image restoration, PDE-constrained · Parameter identification, Primal-dual, Tensor diffusion

1. Introduction

1.1. Overview

One of the original frameworks for image denoising is the Total Variation (TV) problem (in optimization, note that other frameworks are state-of-the-art, in particular in Machine/Deep Learning, which went through CNNs, GANs, and more recently VAEs and Transformers). We refer to [3] for the classical formulation of the TV problem. However, inversion problems, such as denoising, are often ill-posed in Hadamard's sense (existence, uniqueness, and/or stability of the solution not guaranteed). To solve instability, regularization is introduced (see [2]). But, as for Ridge or Lasso methods in classical ML, this method requires manual selection of hyperparameter λ to get the best possible estimator. This can lead to exhausting manual tuning, as well as introduces human potential errors. This is why [1] (the studied article) introduces a trick to avoid this manual selection, ensuring a systematic approach to image denoising. This is particularly relevant when other methods, such as Deep Learning, are completely automated: you get your trained model, feed the noisy image and wait for the result. Automated selection of λ existed but not for the PDE method, which the authors claim to be more robust. Plus, they introduce a new accelerating Primal-Dual algorithm in this paper.

1.2. Problem formulation and model chosen

The problem is formulated in as a PDE-constrained optimization with a Tikhonov regularization term on λ. The denoising problem is formulated as

$$X_0 = X + \mathcal{N}, \tag{1}$$

where X_0 is the observation, X the true image (to be estimated) and $\mathcal{N}$ a Gaussian noise. To reconstruct X, the paper proposes a PDE-constrained optimization approach, where the objective function is given by:

$$\min_{\lambda \in U_{ad}} J(\lambda) = \int_\Omega |X(T,x) - Y_0(x)| dx + \frac{\alpha}{2} \|\nabla \lambda\|^2_{L^2(\Omega)} \tag{2}$$

subject to the following **PDE constraint**:

$$\begin{cases} \frac{\partial X}{\partial t} + \nabla^2 \cdot \left(\lambda \nabla^2 X\right) - \nabla \cdot \left((1-\lambda) D(J_\rho^\sigma(\nabla X)) \nabla X\right) = 0, & \text{in }]0,T[\times\Omega, \\ \langle D(J_\rho^\sigma(\nabla X))\nabla X, \nu\rangle = 0, & \text{on }]0,T[\times\partial\Omega, \\ X(t,x) = 0, & \text{on }]0,T[\times\partial\Omega, \\ X(0,x) = X_0, & \text{on } \Omega, \\ X(T,x) = Y_0(x) & \text{on } \Omega. \end{cases} \tag{3}$$

where $X(T,x)$ is the restored image at time T, Y_0 is a filtered version of the noisy image used as a reference, $\lambda(x)$ is the spatially varying regularization parameter that must be estimated, $D(J_\rho^\sigma(\nabla X))$ is an anisotropic diffusion tensor based on the Weickert model, preserving image structures and the term $\frac{\alpha}{2}\|\nabla\lambda\|^2$ acts as a Tikhonov regularization ensuring stability.

Remark 1. *Note that the model* (3) *is novel in the sense that λ is not a constant scalar hyperparameter but a function to be spatially optimized, meaning that it adapts differently across various regions of the image. It allows the model to perform stronger smoothing in homogeneous regions while preserving details in textured or edge-rich areas.*

Remark 2. *The last equation of* (3) *is set to have enough information to estimate both X and λ. However, one may discuss the choice of the reference Y_0, in this case a bilateral filter. It imposes X to be relatively close to the reference, so less adaptability, but hopefully compensated by the freedom on λ.*

Remark 3. *If the authors introduce a method so that we are no longer forced to choose a constant λ manually, but the choice of hyperparameters is shifted towards the regularization parameter α, and not completely eliminated as their introduction could have led us to think in the first place. Moreover, they fix the choice of Y_0, making it another choice of hyperparameters.*

The PDE model used in this work is a fourth-order diffusion equation that effectively balances denoising and edge preservation. The model consists of two main diffusion operators:

1. A fourth-order isotropic diffusion term, $\nabla^2(\lambda\nabla^2 X)$, ensuring global smoothness across the image.
2. An anisotropic diffusion term, $\nabla \cdot ((1-\lambda)D(J_\rho^\sigma(\nabla X))\nabla X)$, which enhances edge preservation.

2. Primal-Dual formulation, Accelerated Algorithm

To solve this PDE-constrained optimization problem, the paper employs an accelerated Primal-Dual algorithm, which alternates between solving the PDE for X given a current estimate of λ and updating λ using a gradient-based update rule to minimize the cost function.

The Primal-Dual formulation ensures efficient convergence and avoids instability issues that are common in gradient descent-based methods. The problem is formulated as follows:

$$\begin{aligned}\hat{\lambda} &= \arg\min_{\lambda\in U_{ad}} \left\{ \|S(\lambda) - Y_0\|_{L^1(\Omega)} + \frac{\alpha}{2}\|\nabla\lambda\|^2_{L^2(\Omega)} \right\},\\ &= \arg\min_{\lambda\in L^2(\Omega)} \left\{ \|S(\lambda) - Y_0\|_{L^1(\Omega)} + \frac{\alpha}{2}\|\nabla\lambda\|^2_{L^2(\Omega)} + i_{U_{ad}}(\lambda) \right\},\\ &= \arg\min_{\lambda\in L^2(\Omega)} \left\{ F(S(\lambda)) + G(\lambda) \right\}.\end{aligned}$$

The, the dual form is given by:

$$\min_{\lambda\in L^2(\Omega)} \max_{\zeta\in L^\infty(\Omega)\times L^2(\Omega)} \quad G(\lambda) + \langle S(\lambda), \zeta\rangle - F^*(\zeta). \tag{4}$$

We refer to [1] for discussion on the proof of this duality. We see that the dual form is useful to state the problem with convex operators and the authors are indeed able to solve it using the following:

$$\begin{cases} \lambda^{n+1} = (I + \eta\partial G)^{-1}\left(\lambda^n - \eta(S'(\lambda^n))^*\zeta^n\right), \\ \hat{\lambda}^{n+1} = \lambda^{n+1} + \theta(\lambda^{n+1} - \lambda^n), \\ \zeta^{n+1} = (I + \delta\partial F^*)^{-1}\left(\zeta^n + \delta S(\hat{\lambda}^{n+1})\right). \end{cases} \tag{5}$$

This formulation can be solved as the operators can be computed. See [1, Equations 14, 15 and Proposition 1]. We now implement [1, Algorithm 1] and reproduce some of their results.

Remark 4. *We can see that this convex formulation allows to compute both X and λ, while at the same time giving existence and stability of the solution, answering the ill-posed problematic of inverse problems, as discussed in Introduction.*

Remark 5 (Optimization vs. Deep Learning)**.** *Note that this optimization framework is completely interpretable, gives theoretical existence of solutions, and we can understand how it preserves or not shapes and edges (by the obtained λ), a considerable strength compared to implicit and non interpretable methods, such as Deep Learning. Plus, DL is dataset dependent and prone to overfitting, while we may prefer a systematic and generalizable solution to denoising problems, provided in the optimization framework. Finally, the optimization framework is less data consuming and more computationally light compared to DL.*

Remark 6 (This framework vs. "classic" denoising)**.** *As discussed before, this framework allows to compute λ automatically (improving the generalized TV approach). The solution obtained from this paper improves classic denoising (Bilateral in this case but one should be able to choose Gaussian, Median as Y_0...), building on the reference by adapting the strength of regularization spatially ([1, Section 4]).*

3. Experiments

We now reproduce some of the experiments of [1]. To evaluate the performance of methods between each other, we define the structural similarity index (SSIM) and the peak signal-to-noise ratio (PSNR) as:

$$\text{SSIM}(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \tag{6}$$

$$\text{PNSR}(X,Y) = 10\log\left(\frac{m^2}{MSE(X,Y)}\right) \tag{7}$$

where m is the maximum value for a pixel, in our case $m = 1$ if we consider "normalized" images or $m = 255$ before normalization. Moreover, we have μ, σ the means, variance and covariance and C_i positive constants to avoid instability.

Let's now reproduce the experiments of [1]. We use the same parameters, except for the number of iterations N, which is 2000 in the paper but I do not have enough computational power to choose that. Thus, I set $N = 100$. Figure 1 is an example of the beginning of the algorithm, with an image where a Gaussian noise $\sigma = 30$ is introduced. We first reproduce the experiment on the Barbara image (512×512) (see Figure 2), where we obtain $PSNR$: 23.75 $SSIM$: 0.62. Authors obtain better results since they have $N = 2000$. This experiment took more than one hour, showing that the code may be better optimized, but more generally that this method is too expensive.

We conduct further experiments on unseen data (see Figures 3, 4). In all figures, we first observe that the Noisy image exhibits significant grain and degradation compared to the Original. After applying a Bilateral filter, the noise is reduced, but the result remains somewhat blurry, particularly around fine details. By contrast, the proposed PDE-constrained denoising (Restored image) eliminates more noise while preserving sharper edges and textures, illustrating the paper's claim that high-order diffusion operators balance smoothing in homogeneous regions with strong edge retention. Finally, the Spatial λ map reveals how the algorithm automatically adjusts its smoothing strength across the image. However, even if generalization abilities are very good in the Gaussian case (Baboon), if the noise is more general and also caused by low resolution, this PDE method seems to find limits (we see that λ's mapping does not converge and the result is pretty poor) - whereas Deep Learning methods may be able to recreate better de-noised images. Finally, the computational cost of the method and its very complex implementation are to be considered as additional drawbacks revealed by the experiments of the project.

References

[1] Lekbir Afraites et al. "A Weighted Parameter Identification PDE-Constrained Optimization for Inverse Image Denoising Problem". In: *The Visual Computer* 38 (2022), pp. 2883–2898. DOI: 10.1007/s00371-021-02162-x. URL: https://doi.org/10.1007/s00371-021-02162-x.

[2] Kristian Bredies, Karl Kunisch, and Thomas Pock. "Total Generalized Variation". In: *SIAM Journal on Imaging Sciences* 3.3 (2010), pp. 492–526. DOI: 10.1137/090769521. URL: https://doi.org/10.1137/090769521.

[3] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. "Nonlinear Total Variation Based Noise Removal Algorithms". In: *Physica D* 60 (1992), pp. 259–268. DOI: 10.1016/0167-2789(92)90242-F.

Appendix - Experimental Outputs

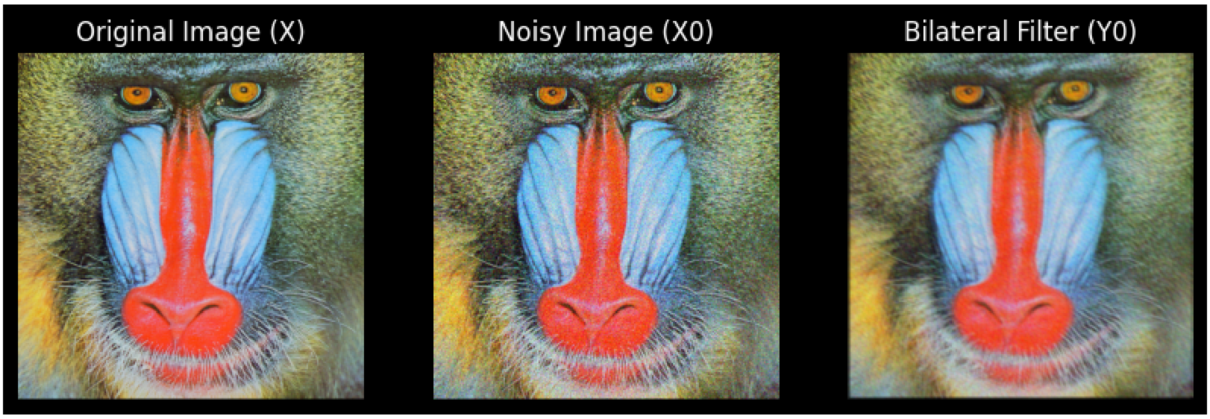

Figure 1: Baboon Image (512×512)

We first reproduce the experiment on the Barbara image (512×512):

Figure 2: Barbara Image and processing (512×512)

Plus, let us conduct the experiments on the Baboon image, and on a noisy Comic image where the noise is not Gaussian (with lower resolution, thus no additional noise is added in this case):

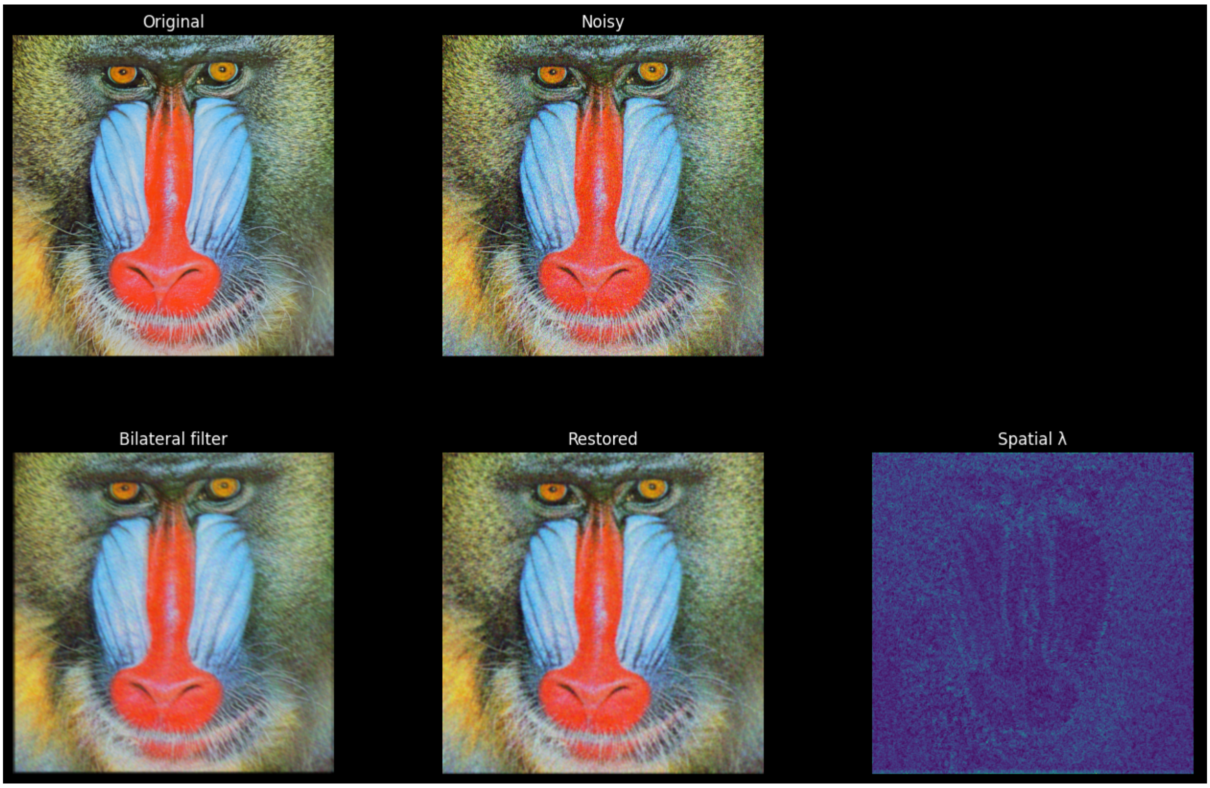

Figure 3: Barbara Image and processing (512×512)

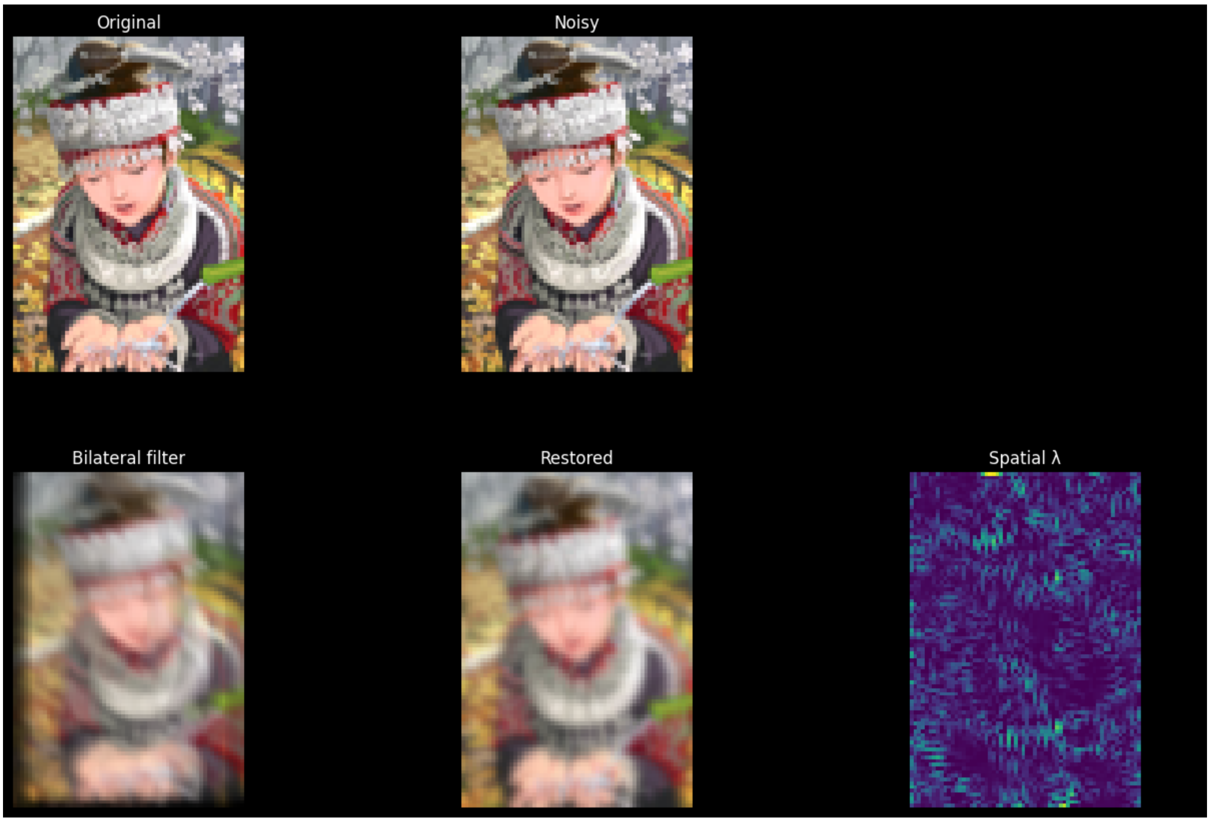

Figure 4: Comic Image and processing (256×256) and $N = 50$