Hyperparameter selection for Discrete Mumford-Shah

Charles-Gérard Lucas, Barbara Pascal, Nelly Pustelnik, Patrice Abry

Hyperparameter selection for Discrete Mumford-Shah *

Charles-Gérard Lucas, Barbara Pascal, Nelly Pustelnik, Patrice Abry †

January 20, 2023

Abstract

This work focuses on a parameter-free joint piecewise smooth image denoising and contour detection. Formulated as the minimization of a discrete Mumford-Shah functional and estimated *via* a theoretically grounded alternating minimization scheme, the bottleneck of such a variational approach lies in the need to fine-tune their hyperparameters, while not having access to ground truth data. To that aim, a Stein-like strategy providing optimal hyperparameters is designed, based on the minimization of an unbiased estimate of the quadratic risk. Efficient and automated minimization of the estimate of the risk crucially relies on an unbiased estimate of the gradient of the risk with respect to hyperparameters. Its practical implementation is performed using a *forward* differentiation of the alternating scheme minimizing the Mumford-Shah functional, requiring exact differentiation of the proximity operators involved. Intensive numerical experiments are performed on synthetic images with different geometry and noise levels, assessing the accuracy and the robustness of the proposed procedure. The resulting *parameter-free piecewise-smooth estimation* and contour detection procedure, not requiring prior image processing expertise nor annotated data, can then be applied to real-world images.

1 Introduction

Context – Image processing is characterized by several key tasks such as image recovery (e.g., debluring and/or denoising), feature extraction, segmentation, and contour detection, to name a few. To provide the user with the requested information, it is standard to perform successively a certain number of these tasks. A first major drawback of cascading tasks, is that important information might be thrown away at each stage. A second key issue is that each task might introduce estimation variance and/or regularization bias, which may accumulate and lead to subsequent errors on the target quantity. Finally, the selection of hyperparameters, e.g., regularization parameters, needs to be performed for each task independently, which might turn sub-optimal overall in minimizing the final error on the output estimate.

The benefit of performing jointly several steps has been illustrated in the context of texture segmentation [9], providing a comparison between a two-step procedure (extract relevant local texture features followed by segmentation) against an original single-step procedure intertwining the estimation of relevant features and the segmentation procedure. Both strategies lead to strongly convex optimization schemes and fair comparisons can be provided by having recourse

*This work is supported by ANR-19-CE48-0009s MULTISC-IN.

†C.-G. Lucas, N. Pustelnik and P. Abry are with ENSL, CNRS, Laboratoire de physique, F-69342 Lyon, France (e-mail: firstname.surname@ens-lyon.fr). B. Pascal is with Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France (e-mail: barbara.pascal@cnrs.fr).

to an automatic hyperparameters selection procedure relying on Stein Unbiased Risk Estimator [13].

Along this line, recent contributions in the image processing literature were dedicated to joint image denoising/restoration and contour detection [7, 14, 15] via biconvex proximal optimization schemes. However, the automatic tuning of hyperparameters in this context has not been dealt with yet: this is the object of the present contribution.

D-MS – This work focuses on a bi-convex formulation, refered as Discrete Mumford-Shah (D-MS) functional, that can trace back to the Mumford-Shah [8] or Geman and Geman functionals [5], aiming to perform joint image denoising and contour detection, which may be written in the discrete variational formulation setting as:

$$\min_{\boldsymbol{u}\in\mathbb{R}^{|\Omega|},\boldsymbol{e}\in\mathbb{R}^{|\mathcal{E}|}} \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{z}\|_2^2 + \beta\|(1-\boldsymbol{e})\odot\mathbf{D}\boldsymbol{u}\|_2^2 + \lambda h(\boldsymbol{e}), \tag{1}$$

where $\boldsymbol{z} = \overline{\boldsymbol{u}} + \sigma\boldsymbol{\zeta} \in \mathbb{R}^{|\Omega|}$ with $\boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}_{|\Omega|}, \mathbf{I}_{|\Omega|})$ denotes the observed degraded image, defined on a grid of pixels Ω such that $|\Omega| = N$, and $\sigma > 0$ is the *known* standard-deviation of the noise. The variable $\boldsymbol{e}$ is a discrete field defined on the lattice $\mathcal{E}$, encapsulating the contour information, whose values are 1 when a contour is detected and 0 otherwise, $\mathbf{D}\colon \mathbb{R}^{|\Omega|} \to \mathbb{R}^{|\mathcal{E}|}$ is a discrete difference operator such that $\mathbf{D}\boldsymbol{u}$ lives on a lattice of contours $\mathcal{E}$, $\odot$ denotes the component-wise product, h denotes a convex separable function enforcing sparsity having its minimum in 0 and such that, for every $\boldsymbol{e} = (e_i)_{1\leq i\leq\mathcal{E}}$, $h(\boldsymbol{e}) = \sum_i h_i(e_i)$. $\beta > 0$ and $\lambda > 0$ are regularization parameters. The minimization is performed with SL-PAM, a nonconvex alternated minimization scheme, with descent parameters genuinely chosen to ensure fast convergence [4] and recalled in Algorithm 4 in the Appendix 4. An exhaustive state-of-the-art regarding variations of the minimization problem (19) is provided in Appendix D.

Hyperparameter selection – The aforementioned procedure for image denoising and contour detection involves *hyperparameters*, e.g. β and λ in (19). To reach satisfactory performance, the fine-tuning of these parameters is crucial. Although central in signal and image processing, this difficult task is still an ongoing challenge, particularly for variational methods.

The difficult problem of the selection of the regularization parameters of the D-MS functional is addressed considering a Stein Unbiased Risk Estimate (SURE), combined with a Finite Difference Monte Carlo (FDMC) strategy making its practical computation tractable. For a detailed state-of-the-art dedicated to hyperparameter selection, the reader could refer to Appendix C. The optimal regularization parameters obtained by minimizing FDMC SURE *via* exhaustive grid search are shown to lead to denoised estimates with high *signal-to-noise ratio* and relevant contours.

Then, to provide a fast procedure selecting the regularization parameters, a Stein Unbiased GrAdient Risk estimate (SUGAR) adapted to D-MS functional (19) is designed, involving the Jacobian of the parametric estimator obtained from (19). Practical implementation of SUGAR requires iterative differentiation of the SL-PAM minimization scheme, for which closed-form formulas are provided. An averaging Monte Carlo strategy is discussed, providing a robust FDMC SUGAR estimator. The resulting procedure compares favorably against exhaustive grid search in terms of *signal-to-noise ratio*, while requiring a significantly smaller computational cost. To the best of our knowledge, the proposed automated D-MS bi-level scheme constitutes a first automated, prior-free and fast discrete Mumford-Shah-like formalism with automated selection of regularization parameters.

Outline – The proposed automated and fast procedure is described in Section 2. Numerical experiments are provided in Section 3.

2 Hyperparameter selection for D-MS

2.1 Stein estimators for D-MS

The minimization of the D-MS functional of Eq. (19) provides both a piecewise smooth image reconstruction, denoted $\widehat{\boldsymbol{u}}(\boldsymbol{z};\beta,\lambda)$ and a set of detected contours, encapsulated into $\widehat{\boldsymbol{e}}(\boldsymbol{z};\beta,\lambda)$, depending on the choice of hyperparameters $\Theta=(\beta,\lambda)\in\mathbb{R}_+\times\mathbb{R}_+$. In such a context, we should ideally minimize a *global* error criterion

$$\widehat{\Theta}\in\underset{\Theta}{\operatorname{Argmin}}\ \mathrm{d}(\widehat{\boldsymbol{x}}(\boldsymbol{z};\Theta),\overline{\boldsymbol{x}}), \tag{2}$$

measuring the ability of the piecewise-smooth image and contour estimates $\widehat{\boldsymbol{x}}(\boldsymbol{z};\Theta)=(\widehat{\boldsymbol{u}}(\boldsymbol{z};\beta,\lambda),\widehat{\boldsymbol{e}}(\boldsymbol{z};\beta,\lambda))$ to approximate the original data $\overline{\boldsymbol{x}}=(\overline{\boldsymbol{u}},\overline{\boldsymbol{e}})$, using a measure of similarity d. If d is chosen to be a quadratic risk, it reads:

$$\mathrm{d}(\widehat{\boldsymbol{x}}(\boldsymbol{z};\Theta),\overline{x})=\mathbb{E}[\|\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)-\overline{\boldsymbol{u}}\|_2^2]+\zeta\mathbb{E}[\|\widehat{\boldsymbol{e}}(\boldsymbol{z};\Theta)-\overline{\boldsymbol{e}}\|_2^2], \tag{3}$$

where $\zeta\geq 0$. However, the degradation model only describes how the observed image $\boldsymbol{z}$ relates to the ground truth image $\overline{\boldsymbol{u}}$, no prior knowledge about how the ground truth contours $\overline{\boldsymbol{e}}$ are affected by the observation noise being assumed. Further, measuring the accuracy of the contours is a tedious task, involving complicated criteria, such as the Jaccard index [6]. For these reasons, in the present work, the quadratic error on which the choice of hyperparameter relies is chosen to be the *quadratic estimation error on the reconstructed image* (i.e. $\zeta=0$).

The present work focuses on a strategy combining Finite Difference approximated differentiation and Monte Carlo averaging, which was first described by [11]. Making use of a Finite Difference step $\epsilon>0$ and a Monte Carlo vector $\boldsymbol{\delta}\in\mathbb{R}^N$ drawn from $\mathcal{N}(\mathbf{0}_N,\mathbf{I}_N)$, Finite Difference Monte Carlo (FDMC) SURE is defined as:

$$\mathrm{SURE}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2):=\|(\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)-\boldsymbol{z})\|_2^2+\frac{2}{\epsilon}\langle\widehat{\boldsymbol{u}}(\boldsymbol{z}+\epsilon\boldsymbol{\delta};\Theta)-\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta),\sigma^2\boldsymbol{\delta}\rangle-\sigma^2N. \tag{4}$$

It involves the denoised image $\widehat{\boldsymbol{u}}(\boldsymbol{z};\beta,\lambda)$, obtained from the minimization of (19), and the standard deviation of the noise σ. Under technical assumptions detailed in Appendix E, the *true* inaccessible quadratic risk estimator (3) when $\zeta=0$ satisfies the following asymptotic unbiasedness property:

$$\lim_{\epsilon\longrightarrow 0}\mathbb{E}[\mathrm{SURE}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2)]=\mathbb{E}[\|\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)-\overline{\boldsymbol{u}}\|_2^2]. \tag{5}$$

Then, the design of a fast *gradient*-based hyperparameter selection strategy providing optimal hyperparameter from the minimization of (25) requires an unbiased estimate $\partial_\Theta\mathrm{SURE}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2)$ of the *gradient* of the quadratic risk with respect to hyperparameters Θ. Such a general procedure is sketched in Algorithm 5 and relies on the FDMC Stein Unbiased GrAdient Risk (SUGAR) estimate defined as:

$$\mathrm{SUGAR}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2)=2\partial_\Theta\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)^*(\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)-\boldsymbol{z})+\frac{2}{\epsilon}\left(\partial_\Theta\widehat{\boldsymbol{u}}(\boldsymbol{z}+\epsilon\boldsymbol{\delta};\Theta)-\partial_\Theta\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)\right)^*\sigma^2\boldsymbol{\delta}, \tag{6}$$

where $\partial_\Theta\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)$ denotes the Jacobian of the parametric estimator $\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)$ with respect to the hyperparameters Θ. The first proposal of such Stein Unbiased GrAdient Risk (SUGAR) estimate was formulated by [3] for i.i.d. Gaussian noise, and then extended in [10] for correlated noise. The main difficulty when it comes to practical implementation is to evaluate the Jacobian matrices.

Algorithm 1 Automated selection of hyperparameters.

Input: Data $\boldsymbol{z}$, ϵ, $\boldsymbol{\delta}$, and true or estimated σ^2.
Initialization: Set $\Theta^{[0]} \in \mathbb{R}^L$.
For $t = 0$ **to** $T_{\max} - 1$ **do**
  Compute $\mathrm{SURE}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z}; \Theta^{[t]} | \sigma^2)$
  Compute $\mathrm{SUGAR}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z}; \Theta^{[t]} | \sigma^2)$
  Update $\Theta^{[t]}$ to $\Theta^{[t+1]}$ *via* a gradient descent step
Output: $\Theta^* = \Theta^{[T_{\max}]}$

2.2 Differentiated SL-PAM

Practical evaluation of the risk and gradient of the risk estimates from Eq. (25) and (6), requires to compute the Jacobian of the D-MS estimator. No closed-form expression being available for $\widehat{\boldsymbol{u}}(\boldsymbol{z}; \beta, \lambda)$, the derivatives are obtained from the iterative differentiation of the recursive scheme of DMS-SLPAM, Algorithm 4 in Appendix D. This strategy raises several technical issues. Indeed, following [3] and [10], the Jacobian matrices $\partial_\Theta \widehat{\boldsymbol{u}}(\boldsymbol{z}; \beta, \lambda)$ and $\partial_\Theta \widehat{\boldsymbol{u}}(\boldsymbol{z} + \epsilon\boldsymbol{\delta}; \beta, \lambda)$ are computed iteratively from a *differentiated* recursive scheme. Particularized to the case of D-MS estimates, the chain differentiation of the SL-PAM scheme (cf. Algorithm 4 in Appendix D) is derived in Algorithm 2. For ease of computation, a specific choice of the step-size $d_k = \eta\beta\|\mathbf{D}\|^2$ involved in the update of the variable $\boldsymbol{e}$ is considered, without inducing any loss of generality.

Algorithm 2 Iterative differentiation of SL-PAM

Input: Data $\widetilde{\boldsymbol{z}} = \{\boldsymbol{z}, \boldsymbol{z} + \epsilon\boldsymbol{\delta}\}$. Set $\Theta = (\beta, \lambda) \in \mathbb{R}_+ \times \mathbb{R}_+$.
Initialization: $\boldsymbol{u}^{[0]} = \widetilde{\boldsymbol{z}}$, $\boldsymbol{e}^{[0]} = \mathbf{1}_{|\mathcal{E}|}$,
$\partial_\Theta \widetilde{\boldsymbol{u}}^{[0]} = \partial_\Theta \boldsymbol{u}^{[0]} = \mathbf{0}_N$, $\partial_\Theta \widetilde{\boldsymbol{e}}^{[0]} = \partial_\Theta \boldsymbol{e}^{[0]} = \mathbf{0}_{|\mathcal{E}|}$.
Set $\gamma > 1$ and $\eta > 0$.
While $|\Psi(\boldsymbol{u}^{[k+1]}, \boldsymbol{e}^{[k+1]}) - \Psi(\boldsymbol{u}^{[k]}, \boldsymbol{e}^{[k]})| > \xi$
  Set $c_k = \gamma\beta\|\mathbf{D}\|^2$ and $d_k = \eta\beta\|\mathbf{D}\|^2$
  $\widetilde{\boldsymbol{u}}^{[k+1]} = \boldsymbol{u}^{(k)} - \frac{1}{c_k}\nabla_{\boldsymbol{u}} g(\boldsymbol{u}^{[k]}, \boldsymbol{e}^{[k]})$
  $\boldsymbol{u}^{[k+1]} = \mathrm{prox}_{\frac{1}{c_k} f(\cdot;\widetilde{\boldsymbol{z}})}\left(\widetilde{\boldsymbol{u}}^{[k+1]}\right)$
  Compute $\partial_\Theta \widetilde{\boldsymbol{u}}^{[k+1]}$ from Eq. (8)
  Compute $\partial_\Theta \boldsymbol{u}^{[k+1]}$ from Eq. (9)
  For all $i \in \{1, \ldots, |\mathcal{E}|\}$
    $$\widetilde{e}_i^{[k]} = \frac{\beta(\mathbf{D}_i \boldsymbol{u}^{[k+1]})^2 + \dfrac{d_k e_i^{[k]}}{2}}{\beta(\mathbf{D}_i \boldsymbol{u}^{[k+1]})^2 + \dfrac{d_k}{2}}$$
    $$e_i^{[k+1]} = \mathrm{prox}_{\frac{\lambda}{2\beta(\mathbf{D}_i \boldsymbol{u}^{[k+1]})^2 + d_k} h_i}(\widetilde{e}_i^{[k]})$$
    Compute $\partial_\Theta \widetilde{e}_i^{[k+1]}$ from Eq. (10)
    Compute $\partial_\Theta e_i^{[k+1]}$ from Eq. (11)

General procedure – The purpose is to differentiate the mapping $\Theta \mapsto (\widehat{\boldsymbol{u}}(\boldsymbol{z}; \Theta), \widehat{\boldsymbol{e}}(\boldsymbol{z}; \Theta))$, where the estimates $(\widehat{\boldsymbol{u}}(\boldsymbol{z}; \Theta), \widehat{\boldsymbol{e}}(\boldsymbol{z}; \Theta))$ are obtained solving (19) for *fixed* $\boldsymbol{z}$.

The recursive *chain differentiation* consists in differentiating step by step DMS-SLPAM each update of which can be written as $\mathbf{v}(\boldsymbol{z}; \Theta) = \Gamma(\mathbf{u}(\boldsymbol{z}; \Theta), \mathbf{e}(\boldsymbol{z}; \Theta), \tau(\Theta))$, where $\mathbf{u} : \mathbb{R}^N \times \mathbb{R}^L \to \mathbb{R}^N$,

$\mathbf{e} : \mathbb{R}^N \times \mathbb{R}^L \to \mathbb{R}^{|\mathcal{E}|}$ and $\tau : \mathbb{R}^L \to \mathbb{R}$ are functions of the observed noisy image $\boldsymbol{z}$ and of the hyperparameters Θ, with respect to which the differentiation is to be performed and $\mathbf{v}(\boldsymbol{z};\Theta) \in \mathcal{K}$, where $\mathcal{K} = \mathbb{R}^N$ when updating $\boldsymbol{u}$ or $\widetilde{\boldsymbol{u}}$, and $\mathcal{K} = \mathbb{R}^{|\mathcal{E}|}$ when updating $\boldsymbol{e}$ or $\widetilde{\boldsymbol{e}}$.

Then, applying the chain rule differentiation principle yields the following partial derivative expression, for every component θ of the hyperparameter vector Θ and for every index $j \in \{1, \dots, \dim(\mathcal{K})\}$:

$$\partial_\theta \mathrm{v}_j = \sum_{\ell=1}^{N} (\partial_{\mathrm{u}_\ell} \Gamma_j)(\partial_\theta \mathrm{u}_\ell) + \sum_{m=1}^{|\mathcal{E}|} (\partial_{\mathrm{e}_m} \Gamma_j)(\partial_\theta \mathrm{e}_m) + (\partial_\tau \Gamma_j)(\partial_\theta \tau) \tag{7}$$

leading to the following closed form expression for $\partial_\Theta \widetilde{\boldsymbol{u}}^{[k+1]}$, $\partial_\Theta \boldsymbol{u}^{[k+1]}$, $\partial_\Theta \widetilde{e}_i^{[k+1]}$ and $\partial_\Theta e_i^{[k+1]}$, for $i \in \{1, \dots, |\mathcal{E}|\}$.

Iterative differentiation of DMS-SLPAM – Applying Formula (7) to each step of DMS-SLPAM leads to Algorithm 2. Proposition 1 provides closed-form expressions of the updates of the Jacobian matrices of the iterates involved in the minimization of a D-MS functional with $h_i = |\cdot|$, thus allowing an easy and direct implementation of FDMC SURE and FDMC SUGAR estimates of Eq. (25) and (6).

Proposition 1. *Considering the D-MS functional* (19) *when* $h_i = |\cdot|$ *and its minimization via SL-PAM Algorithm 3 in Appendix D with* $d_k = \beta\overline{d}$, $\overline{d} = \eta\|\mathbf{D}\|_2^2$, *for every* $\theta \in \{\beta, \lambda\}$*:*

$$\partial_\theta \widetilde{\boldsymbol{u}}^{[k]} = \partial_\theta \boldsymbol{u}^{[k]} - \frac{2\beta}{c_k} \sum_{i=1}^{|\mathcal{E}|} (1 - e_i^{[k]})^2 \ \mathbf{D}_i^* \mathbf{D}_i \partial_\theta \boldsymbol{u}^{[k]} + \frac{4\beta}{c_k} \sum_{i=1}^{|\mathcal{E}|} (1 - e_i^{[k]}) \partial_\theta e_i^{[k]} \ \mathbf{D}_i^* \mathbf{D}_i \boldsymbol{u}^{[k]}, \tag{8}$$

$$\partial_\theta \boldsymbol{u}^{[k+1]} = \frac{c_k}{c_k + 1} \partial_\theta \widetilde{\boldsymbol{u}}^{[k]} + \frac{\widetilde{\boldsymbol{u}}^{[k]} - \boldsymbol{z}}{(\beta\overline{c} + 1)^2} \partial_\theta c_k, \tag{9}$$

where $\partial_\beta c_k = \gamma\|D\|^2$ *and* $\partial_\lambda c_k = 0$*, and for every* $i \in \{1, \dots, |\mathcal{E}|\}$*:*

$$\partial_\theta \widetilde{e}_i^{[k]} = \frac{2\mathbf{D}_i \boldsymbol{u}^{[k+1]} \mathbf{D}_i \partial_\theta \boldsymbol{u}^{[k+1]} \frac{\overline{d}}{2} (1 - e_i^{[k]})}{\left[\left(\mathbf{D}_i \boldsymbol{u}^{[k+1]} \right)^2 + \frac{\overline{d}}{2} \right]^2} + \frac{\frac{\overline{d}}{2} \partial_\theta e_i^{[k]}}{\left(\mathbf{D}_i \boldsymbol{u}^{[k+1]} \right)^2 + \frac{\overline{d}}{2}}, \tag{10}$$

$$\partial_\theta e_i^{[k+1]} = -\partial_{\boldsymbol{u}} \phi_i^{[k+1]} \partial_\theta \boldsymbol{u}^{[k+1]} \frac{\widetilde{e}_i^{[k]}}{|\widetilde{e}_i^{[k]}|} \mathcal{I}_{|\widetilde{e}_i^{[k]}| > \phi_i^{[k+1]}} + \partial_\theta \widetilde{e}_i^{[k]} \mathcal{I}_{|\widetilde{e}_i^{[k]}| > \phi_i^{[k+1]}} - \frac{\partial_\tau \phi_i^{[k+1]} \partial_\theta \tau}{|\widetilde{e}_i^{[k]}|} \widetilde{e}_i^{[k]} \mathcal{I}_{|\widetilde{e}_i^{[k]}| > \phi_i^{[k+1]}}, \tag{11}$$

where

$$\begin{cases} \partial_{\boldsymbol{u}} \phi_i^{[k+1]} \partial_\theta \boldsymbol{u}^{[k+1]} = -\frac{4\tau \mathbf{D}_i \boldsymbol{u}^{[k+1]} \mathbf{D}_i \partial_\theta \boldsymbol{u}^{[k+1]}}{\left[2\left(\mathbf{D}_i \boldsymbol{u}^{[k+1]} \right)^2 + \overline{d} \right]^2}, \\ \partial_\tau \phi_i^{[k+1]} = \frac{1}{2\left(\mathbf{D}_i \boldsymbol{u}^{[k+1]} \right)^2 + \overline{d}}, \\ \partial_\beta \tau = -\frac{\lambda}{\beta^2}, \qquad \partial_\lambda \tau = \frac{1}{\beta}. \end{cases} \tag{12}$$

Proof. The proof is given in Appendix F and the notation $\mathcal{I}$ is defined in Appendix A. □

2.3 Monte Carlo averaging strategy

Following [10], the risk and gradient of the risk FDMC Stein estimators, introduced in Eq. (25) and (6), are defined from *one* realization of the Monte Carlo vector $\boldsymbol{\delta}$. Yet, in the context of a parametric estimator $\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)$ obtained from the minimization of a nonconvex objective functional, such as (19), it can be necessary to go further, and to consider *Monte Carlo averaging* strategies to get more robust risk and gradient of the risk estimates.

The *Monte Carlo averaging* strategy consists in averaging the FDMC Stein estimators of Eq. (25) and (6) over a certain number R of random Monte Carlo vectors $\boldsymbol{\delta}^{(r)} \in \mathbb{R}^N$, independently sampled from the standard Gaussian distribution as stated properly in Definition 1.

Definition 1 (*Monte Carlo averaged* Stein estimators)**.** Let $\boldsymbol{z} = \overline{\boldsymbol{u}} + \sigma\boldsymbol{\zeta} \in \mathbb{R}^{|\Omega|}$ with $\boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}_{|\Omega|}, \mathbf{I}_{|\Omega|})$ and let $\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)$ a parametric estimator of the underlying ground truth $\overline{\boldsymbol{u}}$, depending on some hyperparameters stored in $\Theta \in \mathbb{R}^L$. For $\epsilon > 0$ a Finite Difference step and $\boldsymbol{\Delta} = [\boldsymbol{\delta}^{(1)}, \ldots, \boldsymbol{\delta}^{(R)}]$ a concatenation of independent Monte Carlo vectors sampled from the standard Gaussian distribution. The *Monte Carlo averaged* SURE is defined as

$$\overline{\mathrm{SURE}}^{R}_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z};\Theta) := \frac{1}{R}\sum_{r=1}^{R} \mathrm{SURE}_{\epsilon,\boldsymbol{\delta}^{(r)}}(\boldsymbol{z};\Theta), \tag{13}$$

where $\mathrm{SURE}_{\epsilon,\boldsymbol{\delta}^{(r)}}(\boldsymbol{z};\Theta)$ is the FDMC SURE (25). Similarly, the *Monte Carlo averaged* SUGAR estimator writes

$$\overline{\mathrm{SUGAR}}^{R}_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z};\Theta) := \frac{1}{R}\sum_{r=1}^{R} \mathrm{SUGAR}_{\epsilon,\boldsymbol{\delta}^{(r)}}(\boldsymbol{z};\Theta), \tag{14}$$

involving $\mathrm{SUGAR}_{\epsilon,\boldsymbol{\delta}^{(r)}}(\boldsymbol{z};\Theta)$, the FDMC SUGAR estimate (6).

Proposition 2. *Let $\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)$ an estimator being uniformly Lipschitz w.r.t the observations $\boldsymbol{z}$ and w.r.t the hyperparameters Θ, with a Lipschitz modulus $L_{\widehat{\boldsymbol{u}}(\boldsymbol{z};\cdot)}$ independent of $\boldsymbol{z}$, and satisfying the univocity condition $\widehat{\boldsymbol{u}}(\mathbf{0}_N;\Theta) = \mathbf{0}_N$. For ϵ a infinitesimal positive Finite Difference step and $\boldsymbol{\Delta} = \left[\boldsymbol{\delta}^{(1)}, \ldots, \boldsymbol{\delta}^{(R)}\right]$ a collection of independent standard Gaussian Monte Carlo vectors, the Monte Carlo averaged estimates $\overline{\mathrm{SURE}}^{R}_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z};\Theta)$ and $\overline{\mathrm{SUGAR}}^{R}_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z};\Theta)$ are asymptotically unbiased estimates of respectively the risk and of the gradient of the risk with respect to hyperparameters*

$$\lim_{\epsilon \longrightarrow 0} \mathbb{E}[\overline{\mathrm{SURE}}^{R}_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z};\Theta|\sigma^2)] = Q[\widehat{\boldsymbol{u}}](\Theta) \tag{15}$$

and

$$\lim_{\epsilon \longrightarrow 0} \mathbb{E}[\overline{\mathrm{SUGAR}}^{R}_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z};\Theta|\sigma^2)] = \partial_{\Theta} Q[\widehat{\boldsymbol{u}}](\Theta). \tag{16}$$

Moreover, $\overline{\mathrm{SUGAR}}^{R}_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z};\Theta|\sigma^2)$ is exactly the gradient of $\overline{\mathrm{SURE}}^{R}_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z};\Theta|\sigma^2)$ with respect to the hyperparameters Θ.

Proof. The proof is provided in Appendix G. □

2.4 Averaged SUGAR D-MS

The framework presented in Section 2, combined with Proposition 2, enable us to design an automated strategy to select the D-MS regularization parameters described below and assessed

in Section 3. First, R *independent* Monte Carlo vectors $\boldsymbol{\delta}^{(r)}$ are sampled. The set $\boldsymbol{\Delta} = \{\boldsymbol{\delta}^{(1)}, \ldots, \boldsymbol{\delta}^{(R)}\}$ is kept fixed throughout the procedure. Then, Algorithm 5 with the averaged estimates $\widehat{Q}(\boldsymbol{z}; \Theta|\sigma^2) = \overline{\mathrm{SURE}}^R_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z}; \Theta|\sigma^2)$ and $\partial_\Theta \widehat{Q}(\boldsymbol{z}, \Theta|\sigma^2) = \overline{\mathrm{SUGAR}}^R_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z}; \Theta|\sigma^2)$, defined respectively in Eq. (13) and (14), whose practical implementation is based on Algorithm 2, provides the optimal hyperparameters. The overall procedure is referred to as *Averaged SUGAR D-MS*. Note that, for $R = 1$, $\boldsymbol{\Delta} = \{\boldsymbol{\delta}^{(1)}\}$, and one retrieves the standard SURE and SUGAR estimates. In the case when $R = 1$, the automated hyperparameter strategy is hence referred to as *Standard SUGAR D-MS*.

3 Performance assessment

3.1 Settings

Data – To assess the relevance of SURE (25) in the context of interface detection, as well as the efficiency of the proposed automated minimization making use of the SUGAR proposed in Section 2, systematic experiments are performed on the test data displayed in Fig. 7 and several noise levels are explored, corresponding to $\sigma \in \{0.01, 0.05, 0.1\}$. Additional experiments with other geometries and level of noise are provided in Appendix H.
Algorithmic setup – See Supplementary materials.
Performance criteria – In practice, standard and averaged SURE are compared to the following *quadratic error*: $\mathcal{Q}(\widehat{\boldsymbol{u}}|\overline{\boldsymbol{u}}) := \|\widehat{\boldsymbol{u}} - \overline{\boldsymbol{u}}\|_2^2$. To assess the performance of D-MS denoising with automatically selected hyperparameters, the quality of the reconstruction is quantified by the peak signal-to-noise ratio defined as: $\mathrm{PSNR}(\widehat{\boldsymbol{u}}|\overline{\boldsymbol{u}}) = 20\log_{10}\left(\frac{\|\overline{\boldsymbol{u}}\|}{\|\widehat{\boldsymbol{u}} - \overline{\boldsymbol{u}}\|}\right)$.

3.2 SURE for D-MS

We first illustrate in Fig. 2 the asymptotic unbiasedness of the standard and averaged SURE on the example $\boldsymbol{z}$ displayed in Fig. 7 (top-middle) with noise level $\sigma = 0.05$. To better locate and compare the minima, three level sets of SURE are displayed by the MATLAB function *contour*.

Even though the overall shape of the standard SURE maps are similar to the quadratic error profile, Fig. 2(a-c) shows that the location of the minimum varies significantly with the Monte Carlo vector $\boldsymbol{\delta}^{(r)}$. Averaged SURE also well reproduces the quadratic error map while being more robust to achieve the minimum (cf. Fig. 2(d)).

This first set of experiments illustrate that the proposed averaged SURE reproduces accurately the quadratic risk as expected from Proposition 2

3.3 Comparison between *Standard* and *Averaged SUGAR D-MS*

Fig. 3 investigates the ability of the hyperparameter selection strategies proposed in Section 2.4 for different numbers $R \in \{5, 10, 20\}$ of Monte Carlo vector $\boldsymbol{\delta}^{(r)}$ to achieve the optimal hyperparameters minimizing the quadratic error $\mathcal{Q}(\widehat{\boldsymbol{u}}|\overline{\boldsymbol{u}})$.

The optimal hyperparameters $\Theta^{*(r)} = (\beta^{*(r)}, \lambda^{*(r)})$ reached by the *Standard SUGAR D-MS* are scattered (Fig. 3 left), probably due to a lack of accuracy of the estimator $\mathrm{SUGAR}_{\epsilon,\boldsymbol{\delta}} = \overline{\mathrm{SUGAR}}^{R=1}_{\epsilon,\boldsymbol{\Delta}}$. A first approach to alleviate the variability of the result is to carry out an averaging of R hyperparameters $(\beta^{*(r)}, \lambda^{*(r)})$ obtained by the *Standard SUGAR D-MS* method:

$$\overline{\Theta^*}^R = (\overline{\beta^*}^R, \overline{\lambda^*}^R) = \frac{1}{R}\sum_{r=1}^{R} (\beta^{*(r)}, \lambda^{*(r)}). \tag{17}$$

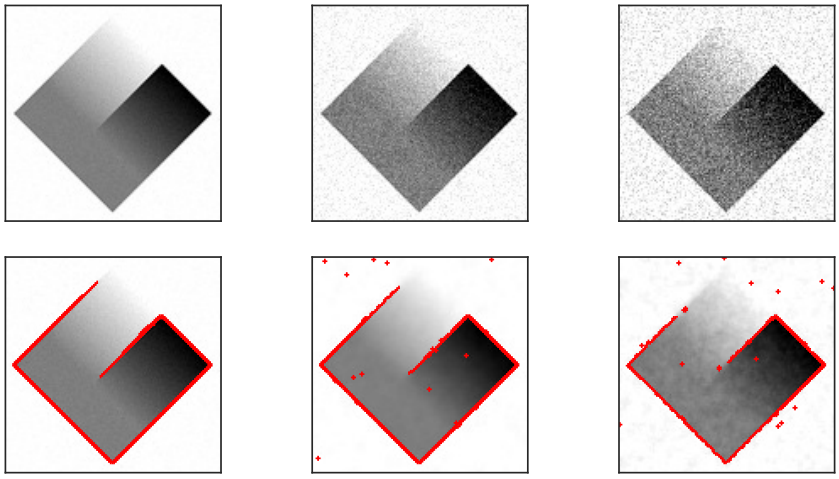


Figure 1: Piecewise smooth grey level image corrupted by i.i.d. Gaussian noise with level $\sigma \in \{0.01, 0.05, 0.1\}$. Associated D-MS estimates $\widehat{\boldsymbol{u}}$ and $\widehat{\boldsymbol{e}}$ (superimposed in red) The D-MS hyperparameters are selected with the proposed *Averaged SUGAR D-MS* (with $R = 5$) using the true standard deviation σ.

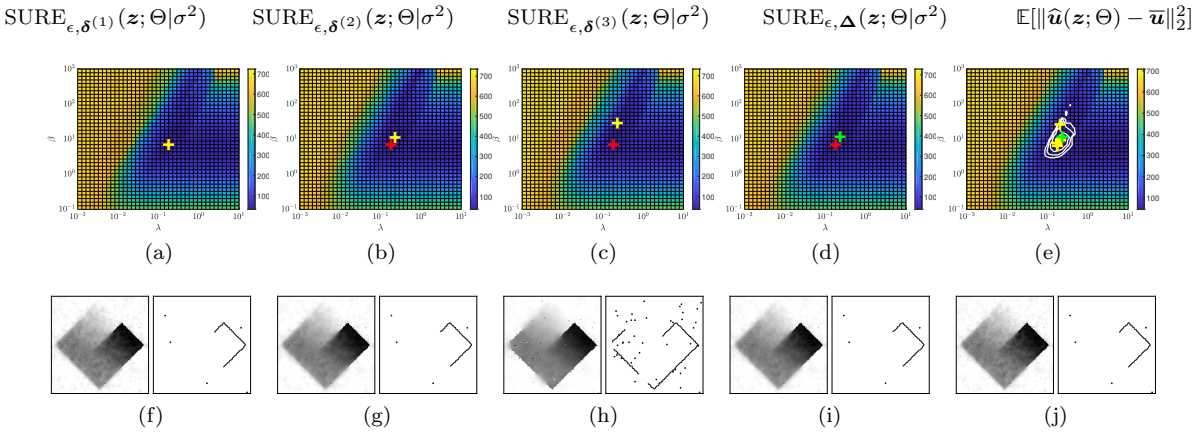


Figure 2: **Comparison between the quadratic error, standard and averaged SURE estimates for D-MS denoising of the image displayed in Fig. 7(middle)**.
1st row – Map on a logarithmic grid of 40×40 hyperparameters $\Theta = (\beta, \lambda)$: (a-c) $\mathrm{SURE}_{\epsilon,\boldsymbol{\delta}^{(r)}}(\boldsymbol{z};\Theta|\sigma^2)$ values for some realizations of the Monte Carlo vector, (d) $\mathrm{SURE}_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z};\Theta|\sigma^2)$ values for $R = 5$ realizations of the Monte Carlo vector and (e) quadratic error $\mathcal{Q}(\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)|\overline{\boldsymbol{u}})$ values with level sets (black lines). **2nd row** – Optimal solutions $\big(\widehat{\mathbf{u}}(\boldsymbol{z};\Theta^{\mathrm{Grid}}), \widehat{\mathbf{e}}(\boldsymbol{z};\Theta^{\mathrm{Grid}})\big)$ obtained from a grid search over each map. The red (resp. yellow and green) cross corresponds to the solution displayed in (j) (resp. (f)-(h) and (i)) associated with the minimum of the quadratic error grid (e) (resp. SURE estimate grids (a)-(c) and (d)).

As it can be observed in Fig. 3 left, this improvement of the method remains unsatisfactory, compared to *Averaged SUGAR D-MS* which reaches more accurate hyperparameters.

The conclusions reached with this set of experiments are twofold: first, we highlight that $R = 5$

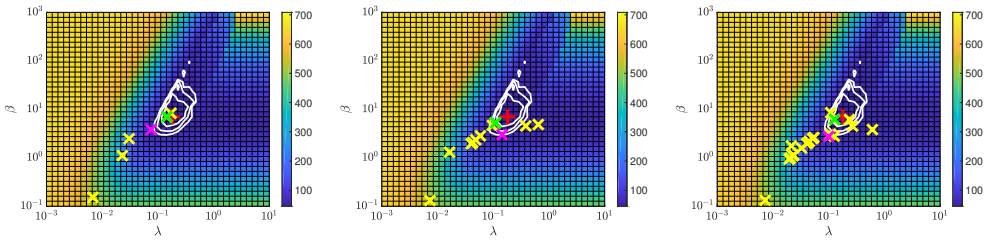


Figure 3: **Impact of the number of realizations R of the Monte Carlo vectors when selecting the hyperparameters with the methods described in Section 2.4**. (left) $R = 5$, (middle) $R = 10$ and (right) $R = 20$. (yellow) *Standard SUGAR D-MS* for different $\boldsymbol{\delta}^{(r)}$ leading to $\Theta^{*(r)}$, (pink) Mean over the R realizations of *Standard SUGAR D-MS* leading to $\overline{\Theta^*}^R$, (green) *Averaged SUGAR D-MS*, (red) optimum obtained by performing a grid search minimization of the quadratic error. For the 3 maps, the background displays the logarithmic grid of 40×40 hyperparameters $\Theta = (\beta, \lambda)$ of quadratic error $\mathcal{Q}(\widehat{\boldsymbol{u}}(\boldsymbol{z}; \Theta)|\overline{\boldsymbol{u}})$ values with level sets (black lines).

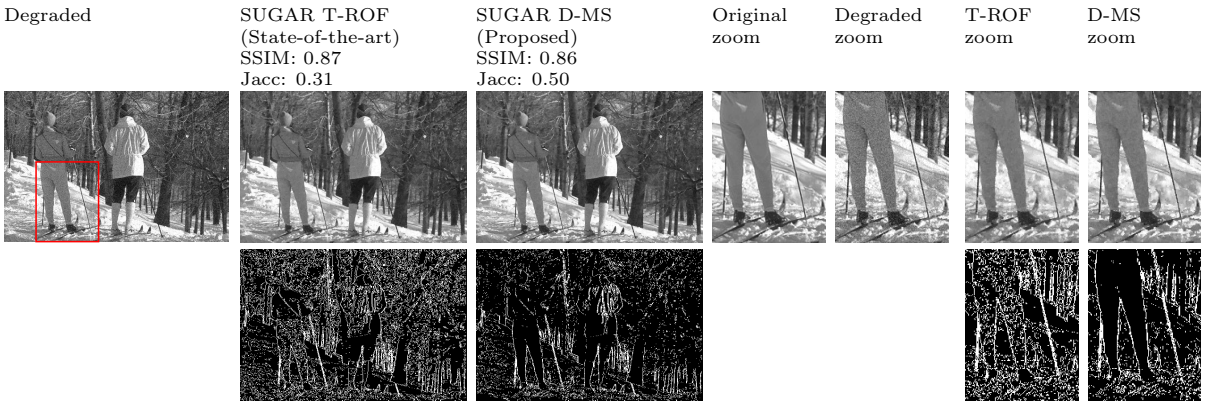


Figure 4: Comparisons between SUGAR T-ROF and SUGAR D-MS for noisy images extracted from the BSD69 dataset [12].

realizations are sufficient to achieve a good estimation of the optimal hyperparameters, second, we note that the proposed automated procedure is 20 times faster compared to exhaustive search, a grid search on averaged SURE requiring 60 minutes of calculation, while *Averaged SUGAR D-MS* requires 3 minutes, when using MATLAB R2018a and an Intel Core i5 processor.

Few estimated images obtained with the fully unsupervised parameter-free *Averaged SUGAR D-MS* are provided in Fig. 7 (2nd row).

3.4 Real-world images

The proposed automated joint denoising and contour detection procedure *Averaged SUGAR D-MS* is evaluated on real-world images extracted from BSD69 dataset [12] degraded with a Gaussian noise with $\sigma = 0.05$. In our experiments we set $R = 5$ and σ has been estimated from noisy data following Eq. (53) in Appendix H.3. Denoised images and contours provided by the proposed data-driven *Averaged SUGAR D-MS* strategy are compared with those yield

by SUGAR T-ROF (a two-step procedure, consisting in, first, a piecewise constant denoising with automated tuning of the regularization parameter [3], followed by an iterative thresholding procedure [2]). In Fig. 8, we can observe that the denoising performance are very close for both procedures (in terms of SSIM, SUGAR T-ROF is slightly better) while the contour detection is significantly improved with the SUGAR D-MS procedure (which is confirmed when computing Jaccard index w.r.t contours obtained from the original image).

4 Conclusion

This work devises a procedure to automatically select the hyperparameters of the D-MS functional allowing to perform simultaneously image denoising and contour detection. This approach is fully unsupervised compared to alternative deep learning strategies such as [1] and reference therein. However, in a future work, it would probably benefit to combine D-MS functional with unfolded deep learning strategies in order to design efficient supervised combined denoising and contour detection approaches.

A MATLAB toolbox implementing the proposed automated image denoising and contour detection procedure is publicly available[1].

A Notations

Let $\mathcal{H}$ a real Hilbert space, and $f : \mathcal{H} \to (+\infty, +\infty]$ a function which is proper, convex, and lower-semicontinuous and $\tau > 0$ a real parameter, the proximity operator of τf at point $\boldsymbol{v} \in \mathcal{H}$ is uniquely defined by $\mathrm{prox}_{\tau f}(\boldsymbol{v}) = \underset{\boldsymbol{u}\in\mathcal{H}}{\arg\min}\, \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{v}\|_2^2 + \tau f(\boldsymbol{v})$. Additionally, let $\mathcal{G}$ be a real Hilbert space and let $\mathcal{A}\colon \mathcal{H} \to \mathcal{G}$ a Lipschitzian map, we denote by $L_{\mathcal{A}} > 0$ the Lipschitz modulus of $\mathcal{A}$, such that, for every $(x, y) \in \mathcal{H}\times\mathcal{H}$, $\|\mathcal{A}(x)-\mathcal{A}(y)\| \leq L_{\mathcal{A}}\|x-y\|$. Further, for every $(x, y) \in \mathbb{R}\times\mathbb{R}$, we denote $\mathcal{I}_{x>y} = 1$ if $x > y$ and 0 otherwise. Finally, $\mathbf{I}_N$ denotes the identity matrix acting on $\mathbb{R}^N$, and $\mathbf{1}_N$ (resp. $\mathbf{0}_N$) is the vector of $\mathbb{R}^N$ containing only ones (resp. zeros).

B State-of-the-art for contour detection in image processing

This work focuses on performing *jointly* piecewise smooth denoising and contour detection on images. In many classical approaches, image reconstruction is embedded into a variational formalism [34, 44], which amounts to find a minimizer of a functional consisting of the sum of a data fidelity term and a prior penalization, i.e.,

$$\underset{\boldsymbol{u}}{\text{minimize}}\, \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{z}\|_2^2 + \gamma p(\mathbf{D}\boldsymbol{u}) \tag{18}$$

where $\gamma > 0$, $\boldsymbol{z} \in \mathbb{R}^{|\Omega|}$ denotes the observed degraded image, defined on a grid of pixels Ω, and $\mathbf{D}\colon \mathbb{R}^{|\Omega|} \to \mathbb{R}^{|\mathcal{E}|}$ is a discrete difference operator such that $\mathbf{D}\boldsymbol{u}$ lives on a lattice of contours $\mathcal{E}$. Appropriate choices of the penalization term p, yield e.g. the Potts functional, when $p = \|\cdot\|_0$, or the Blake and Zisserman functional [21, 23], corresponding to $p(\mathbf{D}\boldsymbol{u}) = \sum_b \min\{\|\mathbf{D}_b\boldsymbol{u}\|_q^q, \chi^q\}$, for some $q \in [1, \infty)$ and $\chi > 0$, with $\mathbf{D}_b$ being associated with several rows of $\mathbf{D}$. In the same vein, considering a convex relaxation of Potts functional, contour detection can be obtained from the minimization of the Rudin-Osher-Fatemi (ROF) functional [41], which favors piecewise constant estimate when considering $p(\mathbf{D}\boldsymbol{u}) = \sum_b \|\mathbf{D}_b\boldsymbol{u}\|_2$. An alternative solution relies on a bi-convex

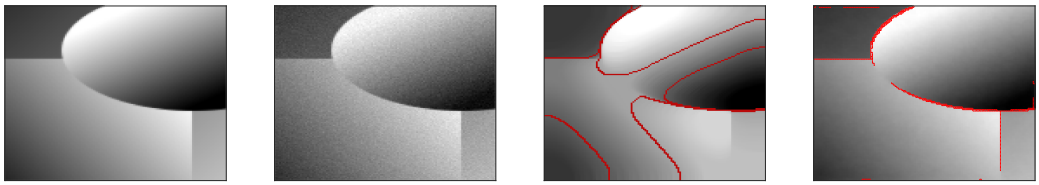
[1] `https://github.com/charlesglucas/sugar_dms`

formulation that can trace back to the Mumford-Shah [8] or Geman and Geman functionals [5], which may be written in the discrete variational formulation setting as:

$$\underset{\boldsymbol{u}\in\mathbb{R}^{|\Omega|},\boldsymbol{e}\in\mathbb{R}^{|\mathcal{E}|}}{\text{minimize}}\ \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{z}\|_2^2+\beta\|(1-\boldsymbol{e})\odot\mathbf{D}\boldsymbol{u}\|_2^2+\lambda h(\boldsymbol{e}), \tag{19}$$

where $\odot$ denotes the component-wise product, h denotes a convex function enforcing sparsity and $\beta>0$ and $\lambda>0$ are regularization parameters. This Discrete Mumford-Shah (D-MS) functional provides a piecewise-smooth reconstructed image $\widehat{\boldsymbol{u}}$ as well as estimated sparse contours $\widehat{\boldsymbol{e}}$.

To achieve segmentation into K regions, Cai and Steidl designed an iterated thresholding strategy [2] applied as a post-processing onto the minimizer of ROF functional. The resulting state-of-the-art two-step procedure, referred to as Thresholded ROF (T-ROF), was proven to be equivalent to minimizing the K-region piecewise constant Mumford-Shah functional. From this thresholded solution, it is then straightforward to identify the contours of the image. However, such an *indirect* contour extraction procedure restricts to *closed* contours. Fig. 5 shows a comparison between D-MS and T-ROF methods on a piecewise smooth image. The Mumford-Shah estimate is piecewise smooth preserving the discontinuities of the image while the ROF estimate is piecewise constant, leading to staircasing effects. We observe that T-ROF erroneously detects interfaces in areas on which the image is piecewise smooth, as opposed to the D-MS whose estimated contour variable is approximately zero everywhere except at the location of the actual signal discontinuity.



(a) Original image (b) Noisy Observation (c) T-ROF (d) D-MS

Figure 5: Comparison of state-of-the-art *convex* variational formulation T-ROF and the studied *non-convex* D-MS performing image denoising and contour extraction. From left to right: (a) Original noise-free piecewise smooth image, (b) Observations $\boldsymbol{z}$ corrupted by an additive Gaussian noise, (c) State-of-the-art ROF piecewise constant estimate and contours derived from thresholding into $K=3$ regions (displayed in red), and (d) Studied D-MS piecewise smooth approximation and estimated contours (displayed in red).

C State-of-the-art for Hyperparameter selection

All aforementioned procedures for image denoising and contour detection involve *hyperparameters*, e.g. β and λ in (19). To reach satisfactory performance, the fine-tuning of these parameters is crucial. Although central in signal and image processing, this difficult task is still an ongoing challenge, particularly for variational methods.

A first class of methods relying on hierarchical Bayesian approaches and has been widely used, both in signal and image processing [19, 27, 37, 45]. The drawbacks of Bayesian methods are that they rapidly become computationally heavy as the model for observed data gets more complicated, and their computational cost increases with the number of hyperparameters to be tuned. For specific 1D denoising problems, efficient hybrid variational/Bayesian strategies can be designed [31].

Several other classes of methods, such as *cross-validation* or Stein Unbiased Risk Estimate (SURE) formulation, can be formulated as a bilevel optimization problem. Cross-validation relies on a given labeled data set composed of noisy samples with their associated ground truth [33, 43]. However, in several real-world applications, such as medical imaging [36] or nonlinear physics problems [39], obtaining a large enough labeled dataset is very challenging, if not impossible. Hence, SURE, initially proposed in [13], has long been favored for its combined simplicity and efficiency. Stein-based hyperparameter strategies rely on an additive Gaussian noise model to design an estimate of the *inaccessible* true risk, defined as the quadratic error between the estimate and ground truth. The major advantage of these approaches is that they do not require to access ground truth. Then, the selection of optimal hyperparameters is done by minimizing SURE and by making use of Finite Difference strategies [42, 46] or/and Monte Carlo averaging [3, 11, 32], to yield tractable and fast implementation of Stein-based risk estimates.

However, the strategy to find the optimal hyperparameters for a specific criterion has a huge impact on the solution both in terms of quality assessment and in terms of computational load. The most standard approach consists in computing a chosen error criterion over a grid of parameters [11, 29, 30], and to select the parameter of the grid for which the error is minimal. Such a grid search procedure suffers from a high computation cost, especially when dealing with $L \geq 2$ regularization parameters. To circumvent this difficulty, efficient automated minimization methods are required. It was early envisionned by Chaux *et al.* [24], who proposed and assessed numerically an empirical descent algorithm for automated choice of regularization parameters, but with no convergence guarantee. A deeper theoretical analysis was then provided by Deledalle *et al.* [3], evidencing sufficient conditions so that Stein Unbiased Risk Estimate is differentiable with respect to hyperparameters, thus enabling to define the Stein Unbiased GrAdient of the Risk (SUGAR) estimator and to provide a practical implementation based on an iterative differentiation strategy. Combining SUGAR with a quasi-Newton descent procedure, a fast algorithm was designed to achieve optimal hyperparameters selection for objective functions of the form (18). This strategy, later extended in [10, 30] for correlated noise, proved its efficiency for texture segmentation [10], piecewise linear signal denoising [39], and in spatial-spectral deconvolution for large multispectral data [17].

D Minimization of the discrete Mumford-Shah functional

The D-MS functional introduced in Eq. (19) being nonconvex, standard proximal algorithms [20, 25, 38] cannot be used directly for its minimization. However, the fact that the functional is separately convex with respect to each variable advocates the use of alternating schemes. Among the vast variety of existing alternating algorithms benefiting from convergence guarantees [4, 18, 22], a numerically efficient procedure for the minimization of D-MS like functionals appears to be the Semi-Linearized Proximal Alternating Minimization (SL-PAM) scheme proposed in [4], whose iterations in the general setting of Problem 1 are recalled in Algorithm 3.

Problem 1 (Nonconvex and nonsmooth minimization)**.** Let $f : \mathbb{R}^{|\Omega|} \to (-\infty, +\infty]$, $h : \mathbb{R}^{|\mathcal{E}|} \to (-\infty, +\infty]$ two proper lower semi-continuous functions and $g : \mathbb{R}^{|\Omega|} \times \mathbb{R}^{|\mathcal{E}|} \to (-\infty + \infty]$ a C^1 function. Let $\lambda > 0$ and $\beta > 0$. We aim to estimate:

$$(\widehat{\boldsymbol{u}}, \widehat{\boldsymbol{e}}) \in \underset{\boldsymbol{u}\in\mathbb{R}^{|\Omega|}, \boldsymbol{e}\in\mathbb{R}^{|\mathcal{E}|}}{\operatorname{Argmin}} \Psi(\boldsymbol{u}, \boldsymbol{e}) := f(\boldsymbol{u}) + \beta g(\boldsymbol{u}, \boldsymbol{e}) + \lambda h(\boldsymbol{e}). \tag{20}$$

The algorithmic scheme SL-PAM (Algorithm 3) is an hybrid version between PAM [18] and PALM [22]. The key ingredient for the efficiency of SL-PAM consists in avoiding the linearization

with respect to the variable $\boldsymbol{e}^{[k]}$, enabling to choose larger descent steps. Under some technical assumptions, such as the existence of a closed-form expressions of the involved proximity operators, the sequence $\left(\boldsymbol{u}^{[k]}, \boldsymbol{e}^{[k]}\right)_{k\in\mathbb{N}}$ converges toward a critical point of $\Psi(\boldsymbol{u}, \boldsymbol{e})$.

Algorithm 3 SL-PAM

Initialization: $\boldsymbol{u}^{[0]} = \boldsymbol{z}$, $\boldsymbol{e}^{[0]} = \mathbf{1}_{|\mathcal{E}|}$, $\gamma > 1$ and $\xi > 0$.
While $|\Psi(\boldsymbol{u}^{[k+1]}, \boldsymbol{e}^{[k+1]}) - \Psi(\boldsymbol{u}^{[k]}, \boldsymbol{e}^{[k]})| > \xi$
- Set $c_k = \gamma L_{\beta\nabla_u g(\cdot, \boldsymbol{e}^{[k]})}$ and $d_k > 0$
- $\widetilde{\boldsymbol{u}}^{[k]} = \boldsymbol{u}^{[k]} - \frac{\beta}{c_k}\nabla_{\boldsymbol{u}} g(\boldsymbol{u}^{[k]}, \boldsymbol{e}^{[k]})$
- $\boldsymbol{u}^{[k+1]} = \operatorname{prox}_{\frac{1}{c_k} f}\left(\widetilde{\boldsymbol{u}}^{[k]}\right)$
- $\boldsymbol{e}^{[k+1]} = \operatorname{prox}_{\frac{1}{d_k}\left(\lambda h + \beta g(\boldsymbol{u}^{[k+1]}, \cdot)\right)}\left(\boldsymbol{e}^{[k]}\right)$

The piecewise smooth image denoising and contour detection strategy defined by (19) and on which this paper focuses corresponds to a particularization of Problem 1. The three terms of the objective function Ψ of Eq. (20) are particularized to

$$\begin{cases} f(\boldsymbol{u}) = \dfrac{1}{2}\|\boldsymbol{u} - \boldsymbol{z}\|_2^2, \\ g(\boldsymbol{u}, \boldsymbol{e}) = \displaystyle\sum_{i=1}^{|\mathcal{E}|} (1 - e_i)^2 (\mathbf{D}_i \boldsymbol{u})^2, \\ h(\boldsymbol{e}) = \displaystyle\sum_{i=1}^{|\mathcal{E}|} h_i(e_i) \end{cases} \tag{21}$$

where, for all $i \in \{1, \ldots, |\mathcal{E}|\}$, $\mathbf{D}_i$ denotes the i^{th}-row of the discrete gradient operator $\mathbf{D}$, and $h_i : \mathbb{R} \mapsto (-\infty, +\infty]$ is a separable proper, lower semi-continuous, and convex function having a proximal operator with known closed-form expression. The iterations of Algorithm 3 specified to the minimization of D-MS lead to Algorithm 4 [4].

For a detailed discussion of the convergence behavior depending on the choice of the descent steps γ and d_k, the reader is referred to [4]. The most efficient setting appears to choose both of them the smallest possible.

E Risk estimation

As previously discussed in introduction, many variational approaches for image restoration and contour detection consists in designing a parametric estimator $\widehat{\boldsymbol{u}}(\boldsymbol{z}; \Theta)$, e.g., defined as a minimizer of (18) or (19), which aims at providing the best possible estimate of a quantity of interest $\overline{\boldsymbol{u}}$ from noisy observations $\boldsymbol{z}$. By construction, the quality of this estimate crucially relies on the precise selection of the hyperparameters Θ, which can be for instance the *regularization parameters* β and λ in D-MS functional (19).

Quadratic risk based parameter selection
The hyperparameters tuning task is commonly formulated as the minimization of the following *quadratic risk*:

$$Q[\widehat{\boldsymbol{u}}](\Theta) = \mathbb{E}[\|\widehat{\boldsymbol{u}}(\boldsymbol{z}; \Theta) - \overline{\boldsymbol{u}}\|_2^2], \tag{22}$$

Algorithm 4 DMS-SLPAM to solve (19)

Input: Data $\boldsymbol{z}$. Set $\beta > 0$, $\lambda > 0$.
Initialization: $\boldsymbol{u}^{[0]} = \boldsymbol{z}$, $\boldsymbol{e}^{[0]} = \mathbf{1}_{|\mathcal{E}|} \in \mathbb{R}^{|\mathcal{E}|}$.
Set $\gamma > 1$ and $\xi > 0$.
While $|\Psi(\boldsymbol{u}^{[k+1]}, \boldsymbol{e}^{[k+1]}) - \Psi(\boldsymbol{u}^{[k]}, \boldsymbol{e}^{[k]})| > \xi$

$$\text{Set } c_k = \gamma\beta\|\mathbf{D}\|^2 \text{ and } d_k > 0.$$
$$\widetilde{\boldsymbol{u}}^{[k]} = \boldsymbol{u}^{[k]} - \tfrac{\beta}{c_k}\nabla_{\boldsymbol{u}} g(\boldsymbol{u}^{[k]}, \boldsymbol{e}^{[k]})$$
$$\boldsymbol{u}^{[k+1]} = \operatorname{prox}_{\frac{1}{c_k} f}(\widetilde{\boldsymbol{u}}^{[k]})$$
For all $i \in \{1, \ldots, |\mathcal{E}|\}$

$$\widetilde{e}_i^{[k]} = \frac{\beta(\mathbf{D}_i \boldsymbol{u}^{[k+1]})^2 + \dfrac{d_k e_i^{[k]}}{2}}{\beta(\mathbf{D}_i \boldsymbol{u}^{[k+1]})^2 + \dfrac{d_k}{2}}$$
$$e_i^{[k+1]} = \operatorname{prox}_{\frac{\lambda}{2\beta(\mathbf{D}_i \boldsymbol{u}^{[k+1]})^2 + d_k} h_i}(\widetilde{e}_i^{[k]})$$

measuring the expected *reconstruction* error made when estimating ground truth $\overline{\boldsymbol{u}}$ by $\widehat{\boldsymbol{u}}(\boldsymbol{z}; \Theta)$. The expectation in Eq. (22) runs over the realizations of the noise corrupting $\boldsymbol{z}$.

In practice, $\overline{\boldsymbol{u}}$ being unknown and the number of observed samples $\boldsymbol{z}$ being limited, if not reduced to one, the exact quadratic risk $Q[\widehat{\boldsymbol{u}}](\Theta)$ of Eq. (22) is not accessible. Thus, the minimization of the quadratic risk $Q[\widehat{\boldsymbol{u}}](\Theta)$ is replaced by the minimization of some *estimate* $\widehat{Q}(\boldsymbol{z}; \Theta|\sigma^2)$ computed from a single noisy sample $\boldsymbol{z}$, not requiring the knowledge of ground truth but only some prior knowledge about the noise, e.g., its standard deviation σ :

$$\widehat{\Theta} \in \operatorname*{Argmin}_{\Theta} \widehat{Q}(\boldsymbol{z}; \Theta|\sigma^2). \tag{23}$$

Then, the design of a fast *gradient*-based hyperparameter selection strategy providing optimal hyperparameters from the minimization of (23) requires an unbiased estimate $\partial_\Theta \widehat{Q}(\boldsymbol{z}; \Theta|\sigma^2)$ of the *gradient* of the quadratic risk with respect to hyperparameters Θ. Such a general procedure is sketched in Algorithm 5.

Algorithm 5 Automated selection of hyperparameters.

Input: Data $\boldsymbol{z}$ and true or estimated σ^2.
Initialization: Set $\Theta^{[0]} \in \mathbb{R}^L$.
For $t = 0$ **to** $T_{\max} - 1$ **do**

Compute $\widehat{Q}(\boldsymbol{z}; \Theta^{[t]}|\sigma^2)$
Compute $\partial_\Theta \widehat{Q}(\boldsymbol{z}; \Theta^{[t]}|\sigma^2)$
Update $\Theta^{[t]}$ to $\Theta^{[t+1]}$ *via* a gradient descent step

Output: $\Theta^* = \Theta^{[T_{\max}]}$

Stein Unbiased Risk Estimate – To address the fact that the ground truth $\overline{\boldsymbol{u}}$ is unknown, the pioneer work of Stein [13] proposed an unbiased estimate of the quadratic risk, based on an i.i.d. Gaussian noise additive model in which the observations are supposed to write

$$\boldsymbol{z} = \overline{\boldsymbol{u}} + \sigma\boldsymbol{\zeta}, \quad \boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}_N, \mathbf{I}_N) \tag{24}$$

with $N = |\Omega|$ is the number of pixels and σ^2 the *known* variance of the noise. Then, under integrability and regularity assumptions, together with the observation model (24), the so-called Stein Unbiased Risk Estimator (SURE) was derived in [13], and has then been intensively used in signal and image processing [9, 16, 24, 39, 40]. In most applications, the original Stein estimator is not usable directly and further strategies are necessary to yield a practical estimator. The present work focuses on a strategy combining Finite Difference approximated differentiation and Monte Carlo averaging, which was first described by [11]. Making use of a Finite Difference step $\epsilon > 0$ and a Monte Carlo vector $\boldsymbol{\delta} \in \mathbb{R}^N$ drawn from $\mathcal{N}(\mathbf{0}_N, \mathbf{I}_N)$, Finite Difference Monte Carlo (FDMC) SURE is defined as:

$$\mathrm{SURE}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2) := \|(\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta) - \boldsymbol{z})\|_2^2 + \frac{2}{\epsilon}\langle \widehat{\boldsymbol{u}}(\boldsymbol{z}+\epsilon\boldsymbol{\delta};\Theta) - \widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta), \sigma^2\boldsymbol{\delta}\rangle - \sigma^2 N, \tag{25}$$

Under the Lipschitzianity with respect to $\boldsymbol{z}$ of $\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)$ and the natural unambiguity property $\widehat{\boldsymbol{u}}(\mathbf{0}_N,\Theta) = \mathbf{0}_N$, the *true* inaccessible quadratic risk estimator (22) satisfies the following asymptotic unbiasedness property:

$$\lim_{\epsilon \longrightarrow 0} \mathbb{E}[\mathrm{SURE}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2)] = Q[\widehat{\boldsymbol{u}}](\Theta), \tag{26}$$

where the expectation is to be understood on both the realizations of the observation noise $\boldsymbol{\zeta}$ appearing in Eq. (24), and the realizations of the Monte Carlo vector $\boldsymbol{\delta}$. Eq. (26) ensures that, for small enough Finite Difference step ϵ, and provided that N is large enough so that the Monte Carlo strategy is relevant, a minimizer of $\mathrm{SURE}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2)$ is an approximately optimal set of hyperparameters in terms of quadratic risk.

Risk estimate minimization – The gradient-based strategy sketched at Algorithm 5 when

$$\widehat{Q}(\boldsymbol{z};\Theta|\sigma^2) = \mathrm{SURE}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2) \tag{27}$$

relies on the FDMC Stein Unbiased GrAdient Risk (SUGAR) estimate defined as:

$$\mathrm{SUGAR}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2) = 2\partial_\Theta \widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)^*(\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta) - \boldsymbol{z}) + \frac{2}{\epsilon}\left(\partial_\Theta \widehat{\boldsymbol{u}}(\boldsymbol{z}+\epsilon\boldsymbol{\delta};\Theta) - \partial_\Theta \widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)\right)^* \sigma^2\boldsymbol{\delta}, \tag{28}$$

where $\partial_\Theta \widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)$ denotes the Jacobian of the parametric estimator $\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)$ with respect to the hyperparameters Θ. The first proposal of such Stein Unbiased GrAdient Risk (SUGAR) estimate was formulated by [3] for i.i.d. Gaussian noise, and then extended in [10] for correlated noise. The main difficulty when it comes to practical implementation is to evaluate the Jacobian matrices. In [3, 10], the authors proposed an efficient implementation when $\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)$ is estimated from the resolution of a convex minimization problem of the form (18) while in this contribution we extend it in the context of interface detection involving a minimization problem such as (19) solved with SL-PAM described in Section D.

Under technical assumptions such as Lipschitzianity of $\widehat{\boldsymbol{u}}(\boldsymbol{z};\Theta)$ with respect to Θ and $\boldsymbol{z}$, it has been proved in [3] that the quadratic risk estimator (25) is weakly differentiable with respect to Θ and its gradient is exactly the gradient estimator recalled in (28), i.e.,

$$\partial_\Theta \mathrm{SURE}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2) = \mathrm{SUGAR}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2). \tag{29}$$

Eq. (29) ensures that the gradient estimate $\mathrm{SUGAR}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2)$ is indeed the gradient of the quadratic risk estimate $\mathrm{SURE}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2)$ with respect to hyperparameters Θ, justifying the use of the gradient descent approach of Algorithm 5 to solve a particular instance of Problem (23)

when $\widehat{Q}(\boldsymbol{z};\Theta|\sigma^2)$ is defined by (27). Additionally, FDMC SUGAR estimator introduced in (28) is an asymptotically unbiased estimator of the gradient of the *true* quadratic risk, i.e.

$$\lim_{\epsilon \longrightarrow 0} \mathbb{E}[\mathrm{SUGAR}_{\epsilon,\boldsymbol{\delta}}(\boldsymbol{z};\Theta|\sigma^2)] = \partial_\Theta Q[\widehat{\boldsymbol{u}}](\Theta), \tag{30}$$

where $\partial_\Theta Q[\widehat{\boldsymbol{u}}](\Theta)$ is the *true* inaccessible gradient of quadratic risk with respect to hyperparameters Θ, and the expectation is to be understood on both the realizations of the observation noise $\boldsymbol{\zeta}$ appearing in Eq. (24) and the realizations of the Monte Carlo vector $\boldsymbol{\delta}$. The asymptotic unbiasedness of the gradient estimate ensures that the risk profile around its minimum is well enough reproduced by Stein-like estimates so that Algorithm 5 can be reasonably supposed to output a good approximation of the *true* optimal hyperparameters.

F Iterative differentiation of SL-PAM for D-MS

F.1 Update of $\partial_\theta \tilde{\boldsymbol{u}}^{[k]}$

For the function g given in Eq. (21), the update rule of $\widetilde{\boldsymbol{u}}^{[k]}$ reads:

$$\begin{aligned} \widetilde{\boldsymbol{u}}^{[k]} &= \boldsymbol{u}^{[k]} - \frac{\beta}{c_k} \nabla_u g(\boldsymbol{u}^{[k]}, \boldsymbol{e}^{[k]}) \\ &= \boldsymbol{u}^{[k]} - \frac{2\beta}{c_k} \sum_{i=1}^{|\mathcal{E}|} (1 - e_i^{[k]})^2 \ \mathbf{D}_i^* \mathbf{D}_i \boldsymbol{u}^{[k]}. \end{aligned} \tag{31}$$

The update of $\tilde{\boldsymbol{u}}^{[k]}$ can be written:

$$\tilde{\boldsymbol{u}}^{[k]} = \Gamma(\boldsymbol{u}^{[k]}, \boldsymbol{e}^{[k]}, \tau^{[k]}), \tag{32}$$

where

$$\begin{cases} \Gamma(\mathbf{u}, \mathbf{e}, \tau) = \mathbf{u} - \tau \sum_{i=1}^{|\mathcal{E}|} (1 - \mathrm{e}_i)^2 \ \mathbf{D}_i^* \mathbf{D}_i \mathbf{u}, \\ \tau^{[k]} = \frac{2\beta}{c_k}. \end{cases} \tag{33}$$

The derivative $\partial_\theta \mathbf{v}$ of $\mathbf{v} = \Gamma(\mathbf{u}, \mathbf{e}, \tau)$, for $\theta \in \{\beta, \lambda\}$, is:

$$\begin{aligned} \partial_\theta \mathbf{v} =& \partial_\theta \mathbf{u} - \tau \sum_{i=1}^{|\mathcal{E}|} (1 - \mathrm{e}_i)^2 \ \mathbf{D}_i^* \mathbf{D}_i \partial_\theta \mathbf{u} \\ &+ 2\tau \sum_{i=1}^{|\mathcal{E}|} (1 - \mathrm{e}_i) \partial_\theta \mathrm{e}_i \ \mathbf{D}_i^* \mathbf{D}_i \mathbf{u} \\ &- \partial_\theta \tau \sum_{i=1}^{|\mathcal{E}|} (1 - \mathrm{e}_i)^2 \mathbf{D}_i^* \mathbf{D}_i \mathbf{u}, \end{aligned} \tag{34}$$

and, since $c_k = \beta\gamma\|D\|^2$, for $\theta \in \{\beta, \lambda\}$,

$$\partial_\theta \tau^{[k]} = \partial_\theta \left(\frac{2\beta}{c_k} \right) = \partial_\theta \left(\frac{2}{\gamma \|D\|^2} \right) = 0. \tag{35}$$

F.2 Update of $\partial_\theta \boldsymbol{u}^{[k+1]}$

The function f given in Eq. (21) has a proximal operator with a closed form expression. Thus the update rule of $\boldsymbol{u}^{[k]}$ can be explicitely expressed as follows:

$$\boldsymbol{u}^{[k+1]} = \operatorname{prox}_{\frac{1}{c_k} f}(\widetilde{\boldsymbol{u}}^{[k]}) = \frac{c_k \widetilde{\boldsymbol{u}}^{[k]} + \boldsymbol{z}}{c_k + 1}. \tag{36}$$

For the update of $\boldsymbol{u}^{[k+1]}$, we thus have:

$$\boldsymbol{u}^{[k+1]} = \Gamma(\tilde{\boldsymbol{u}}^{[k]}, \mathbf{0}_{|\mathcal{E}|}, \tau^{[k]}), \tag{37}$$

where

$$\begin{cases} \Gamma(\mathbf{u}, \mathbf{e}, \tau) = \frac{\tau \mathbf{u} + \boldsymbol{z}}{\tau + 1}, \\ \tau^{[k]} = c_k. \end{cases} \tag{38}$$

The derivative $\partial_\theta \mathbf{v}$ of $\mathbf{v} = \Gamma(\mathbf{v}, \mathbf{e}, \tau)$, for $\theta \in \{\beta, \lambda\}$, is:

$$\partial_\theta \mathbf{v} = \frac{\tau}{\tau + 1} \partial_\theta \mathbf{u} + \frac{\mathbf{u} - \boldsymbol{z}}{(\tau + 1)^2} \partial_\theta \tau, \tag{39}$$

and $\partial_\beta \tau^{[k]} = \gamma \|\mathbf{D}\|^2$ and $\partial_\lambda \tau^{[k]} = 0$.

F.3 Update of $\partial_\theta \tilde{\boldsymbol{e}}^{[k]}$

In Algorithm 4, the parameter for the update of $\boldsymbol{e}^{[k+1]}$ is set to $d_k = \beta \overline{d}$ where $\overline{d} = \eta \|\mathbf{D}\|_2^2$. This choice is discussed in [4] and gives good numerical results for some values of η. This setting simplifies the computation of the derivatives due to the linear dependance of d_k with β. Indeed, the update rule of $\widetilde{e}_i^{[k]}$, for every $i \in \{1, \dots, |\mathcal{E}|\}$, can be rewritten as follows:

$$\widetilde{e}_i^{[k]} = \frac{(\mathbf{D}_i \boldsymbol{u}^{[k+1]})^2 + \frac{\overline{d} e_i^{[k]}}{2}}{(\mathbf{D}_i \boldsymbol{u}^{[k+1]})^2 + \frac{\overline{d}}{2}}. \tag{40}$$

Thus, for every $i \in \{1, \dots, |\mathcal{E}|\}$,

$$\tilde{e}_i^{[k]} = \Gamma_i(\boldsymbol{u}^{[k+1]}, \boldsymbol{e}^{[k]}, \tau^{[k]}), \tag{41}$$

where

$$\begin{cases} \Gamma_i(\mathbf{u}, \mathbf{e}, \tau) = \frac{(\mathbf{D}_i \mathbf{u})^2 + \frac{\tau}{2} \mathrm{e}_i}{(\mathbf{D}_i \mathbf{u})^2 + \frac{\tau}{2}}, \\ \tau^{[k]} = \overline{d}. \end{cases} \tag{42}$$

The derivative $\partial_\theta \mathbf{v}$ of $\mathbf{v} = \Gamma(\mathbf{u}, \mathbf{e}, \tau)$, for $\theta \in \{\beta, \lambda\}$ and for $i \in \{1, \dots, |\mathcal{E}|\}$, is:

$$\begin{aligned} \partial_\theta \mathrm{v}_i = & \frac{2 \mathbf{D}_i \mathbf{u} \mathbf{D}_i \partial_\theta \mathbf{u}}{(\mathbf{D}_i \mathbf{u})^2 + \frac{\tau}{2}} - \frac{2 \mathbf{D}_i \mathbf{u} \mathbf{D}_i \partial_\theta \mathbf{u} \left[(\mathbf{D}_i \mathbf{u})^2 + \frac{\tau}{2} \mathrm{e}_i \right]}{\left[(\mathbf{D}_i \mathbf{u})^2 + \frac{\tau}{2} \right]^2} \\ & + \frac{\frac{\tau}{2} \partial_\theta \mathrm{e}_i}{(\mathbf{D}_i \mathbf{u})^2 + \frac{\tau}{2}} \\ & + \frac{\frac{\partial_\theta \tau}{2} \mathrm{e}_i}{(\mathbf{D}_i \mathbf{u})^2 + \frac{\tau}{2}} - \frac{[(\mathbf{D}_i \mathbf{u})^2 + \frac{\tau}{2} \mathrm{e}_i] \frac{\partial_\theta \tau}{2}}{\left[(\mathbf{D}_i \mathbf{u})^2 + \frac{\tau}{2} \right]^2}, \end{aligned} \tag{43}$$

and $\partial_\theta \tau^{[k]} = 0$, which yields to the result in [35, Proposition 2].

F.4 Update of $\partial_\theta \boldsymbol{e}^{[k+1]}$

For the update of $e_i^{[k+1]}$, for every $i \in \{1, \dots, |\mathcal{E}|\}$, the fonction h in Eq. (21) is chosen with $h_i = |.|$. This latter function corresponds to the common ℓ_1-norm penalization of the contour. The setting $d_k = \beta \overline{d}$ simplifies this update which now reads:

$$e_i^{[k+1]} = \mathrm{prox}_{\phi_i^{[k+1]}|.|}(\widetilde{e}_i^{[k]}), \quad \phi_i^{[k+1]} = \phi_i(\boldsymbol{u}^{[k+1]}; \tau), \tag{44}$$

where

$$\begin{cases} \phi_i(\mathbf{u}, \tau) = \frac{\tau}{\left[2(\mathbf{D}_i\mathbf{u})^2 + \overline{d}\right]}, \quad \tau = \frac{\lambda}{\beta}, \\ \mathrm{prox}_{\phi_i(\mathbf{u},\tau)|.|}(\mathrm{e}_i) = \max(0, 1 - \frac{\phi_i(\mathbf{u};\tau)}{|\mathrm{e}_i|})\mathrm{e}_i. \end{cases} \tag{45}$$

For every $i \in \{1, \dots, |\mathcal{E}|\}$,

$$e_i^{[k+1]} = \Gamma_i(\boldsymbol{u}^{[k+1]}, \tilde{\boldsymbol{e}}^{[k]}, \tau^{[k]}), \tag{46}$$

where

$$\begin{cases} \Gamma_i(\mathbf{u}, \mathbf{e}, \tau) = \mathrm{prox}_{\phi_i(\mathbf{u};\tau)|.|}(\mathrm{e}_i), \\ \tau^{[k]} = \frac{\lambda}{\beta}. \end{cases} \tag{47}$$

The derivative $\partial_\theta \mathbf{v}$ of $\mathbf{v} = \Gamma(\mathbf{u}, \mathbf{e}, \tau)$, for $\theta \in \{\beta, \lambda\}$ and for $i \in \{1, \dots, |\mathcal{E}|\}$, is:

$$\begin{aligned} \partial_\theta \mathrm{v}_i = &- \partial_{\mathbf{u}}\phi_i \partial_\theta \mathbf{u} \frac{\mathrm{e}_i}{|\mathrm{e}_i|} \mathcal{I}_{|\mathrm{e}_i| > \phi_i(\mathbf{u},\tau)} + \partial_\theta \mathrm{e}_i \mathcal{I}_{|\mathrm{e}_i| > \phi_i(\mathbf{u},\tau)} \\ &- \frac{\partial_\theta \tau}{\left[2\left(\mathbf{D}_i\mathbf{u}\right)^2 + \overline{d}\right]} \frac{\mathrm{e}_i}{|\mathrm{e}_i|} \mathcal{I}_{|\mathrm{e}_i| > \phi_i(\mathbf{u},\tau)}, \end{aligned} \tag{48}$$

with Jacobian matrices product

$$\partial_{\mathbf{u}}\phi_i \partial_\theta \mathbf{u} = -\frac{\tau}{\left[2\left(\mathbf{D}_i\mathbf{u}\right)^2 + \overline{d}\right]^2} \left(4\mathbf{D}_i\mathbf{u}\mathbf{D}_i\partial_\theta\mathbf{u}\right), \tag{49}$$

and $\partial_\beta \tau^{[k]} = \frac{1}{\beta}$ and $\partial_\lambda \tau^{[k]} = -\frac{\lambda}{\beta^2}$.

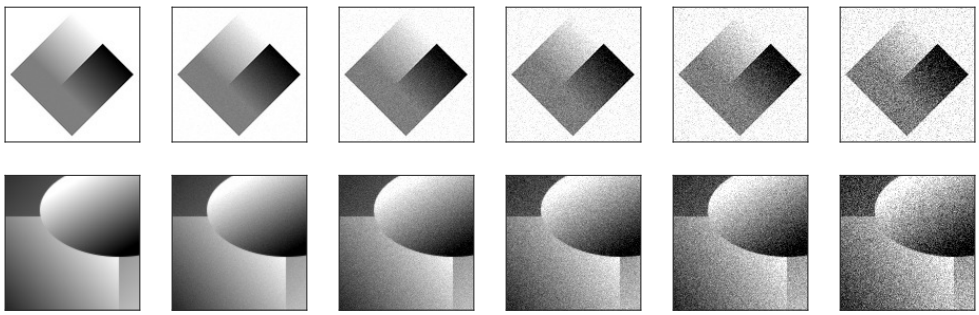
G Proof of Proposition 2

For each Monte Carlo vector $\boldsymbol{\delta}^{(r)}$, the FDMC $\mathrm{SURE}_{\epsilon,\boldsymbol{\delta}^{(r)}}$ and $\mathrm{SUGAR}_{\epsilon,\boldsymbol{\delta}^{(r)}}$ estimates, defined at Eq. (25) and (28) are asymptotically unbiased and $\mathrm{SUGAR}_{\epsilon,\boldsymbol{\delta}^{(r)}}$, is the derivative of $\mathrm{SURE}_{\epsilon,\boldsymbol{\delta}^{(r)}}$, w.r.t. Θ. Then, by linearity of both the limit $\lim_{\epsilon\to 0}$ and the summation over the R Monte Carlo vectors, the *Monte Carlo averaged* estimates $\overline{\mathrm{SURE}}^R_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z}; \Theta|\sigma^2)$ and $\overline{\mathrm{SUGAR}}^R_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z}; \Theta|\sigma^2)$ are also unbiased and $\overline{\mathrm{SUGAR}}^R_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z}; \Theta|\sigma^2)$ is the derivative of $\overline{\mathrm{SURE}}^R_{\epsilon,\boldsymbol{\Delta}}(\boldsymbol{z}; \Theta|\sigma^2)$ w.r.t. Θ.

H Additional experiments

H.1 Algorithmic setup

SL-PAM – The minimization of the D-MS functional (19) providing estimates of both the piecewise smooth image and its salient contours, is performed running Algorithm 4. The stopping criterion, based on the objective function increments, is set to $\xi = 10^{-4}$, while the descent steps



(a) $\sigma = 0$ (b) $\sigma = 0.01$ (c) $\sigma = 0.03$ (d) $\sigma = 0.05$ (e) $\sigma = 0.07$ (f) $\sigma = 0.1$

Figure 6: Piecewise smooth grey level images ($\sigma = 0$) corrupted by i.i.d. Gaussian noise with level $\sigma \in \{0.01, 0.03, 0.05, 0.07, 0.1\}$.

are tuned manually so as to obtain the fastest convergence, leading to $\gamma = 1.01$ and $d_k = \eta\beta\|\mathbf{D}\|_2^2$ with $\eta = 1.01 \times 10^{-3}$ following [4].

Stein estimators – FDMC SURE (25) is computed with a Finite Difference step

$$\epsilon = 2\frac{\sigma}{N^\alpha}, \quad 0 < \alpha < 1 \tag{50}$$

where σ is the standard deviation of the noise on the observed image $\boldsymbol{z} \in \mathbb{R}^N$. Formula (50) derives from a heuristic reasoning developed in [3], in the context of ℓ_1-norm penalization. The dependency of the Finite Difference step on the size of the data is controlled via the exponent α, which is fixed at $\alpha = 0.3$ for all the numerical simulations. In the systematic numerical experiments, four values of the number R of realizations of the Monte Carlo vector $\boldsymbol{\delta}^{(r)}$ are envisioned and systematically compared: $R \in \{1, 5, 10, 20\}$.

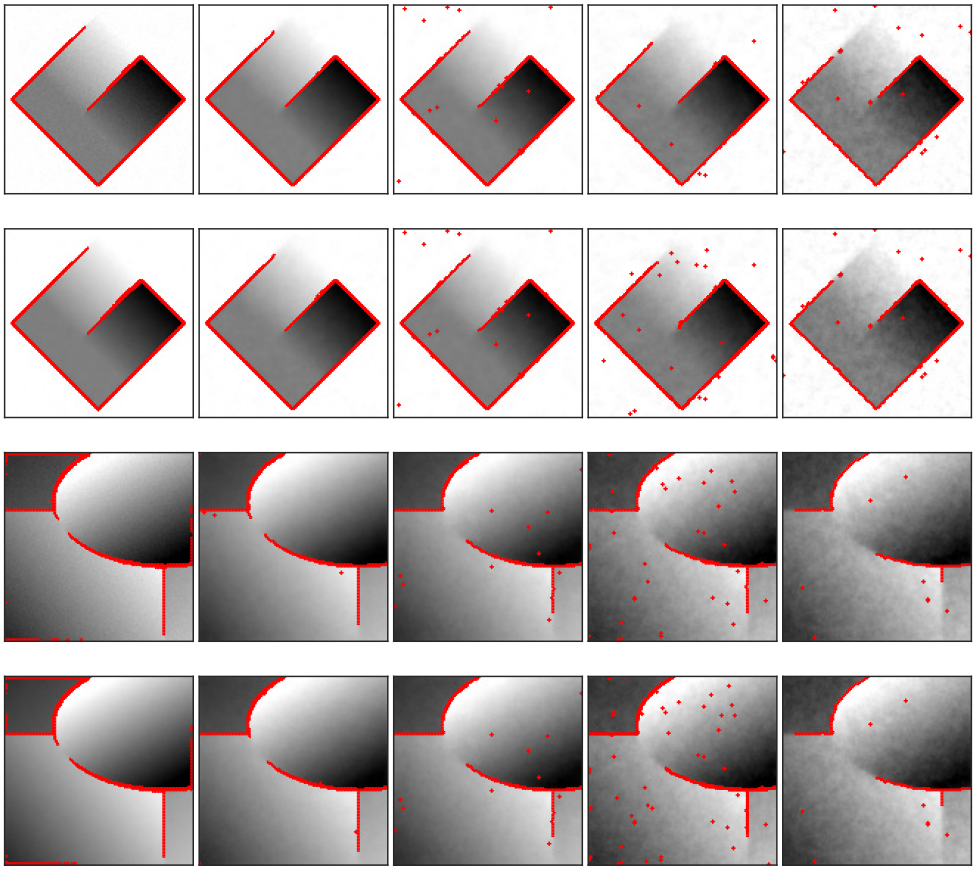
BFGS algorithm – To perform the risk minimization described in Algorithm 5 for different choices of $\widehat{Q}(\boldsymbol{z}; \Theta\,|\sigma^2)$ and $\partial_\Theta \widehat{Q}(\boldsymbol{z}; \Theta\,|\sigma^2)$ we used the GRadient-based Algorithm for Non-Smooth Optimization, implemented in GRANSO toolbox[2], consisting of the low memory BFGS quasi-Newton algorithm proposed in [26] with box constraints, enabling to enforce positivity of β and λ. The maximal number of iterations of BFGS Algorithm 5 is set to $T_{\max} = 20$, while the stopping criterion on the gradient norm is set to 10^{-8}. Further, it is well-documented that the initialization of quasi-Newton algorithms might drastically impact their convergence. Hence, we propose a model-based strategy for initializing Algorithm 5. Inspired from the initialization strategies proposed in [3, 10], hyperparameters $\Theta = (\beta, \lambda)$ are initialized as

$$\beta^{(0)} = \frac{N\sigma^2}{4\|\mathbf{D}\boldsymbol{z}\|_2^2} \quad \text{and} \quad \lambda^{(0)} = \frac{\beta^{(0)}\|\mathbf{D}\boldsymbol{z}\|_2^2}{2N}, \tag{51}$$

while, for $\kappa = 0.9$, the initial approximated inverse Hessian involved in the BFGS strategy is set to

$$\mathrm{H}^{(0)} = \mathrm{diag}\left(\left|\frac{\kappa\beta^{(0)}}{\partial_\beta \widehat{Q}(\boldsymbol{z}; \Theta^{(0)}|\sigma^2)}\right|, \left|\frac{\kappa\lambda^{(0)}}{\partial_\lambda \widehat{Q}(\boldsymbol{z}; \Theta^{(0)}\ |\sigma^2)}\right|\right). \tag{52}$$

[2] `http://www.timmitchell.com/software/GRANSO/`



(a) $\sigma = 0.01$ (b) $\sigma = 0.03$ (c) $\sigma = 0.05$ (d) $\sigma = 0.07$ (e) $\sigma = 0.1$

Figure 7: D-MS estimates $\widehat{\boldsymbol{u}}$ and $\widehat{\boldsymbol{e}}$ (superimposed in red) of piecewise smooth grey level images corrupted by i.i.d. Gaussian noise with noise level σ displayed in Fig. 6. The D-MS hyperparameters are selected with the proposed *Averaged SUGAR D-MS* using either the true standard deviation σ (first and third rows) or the estimated standard deviation $\widehat{\sigma}$ (second and fourth rows).

H.2 Performance w.r.t noise level

We now focus on the *Averaged SUGAR D-MS* for $R = 5$ and assess its performance for the different geometries and noise levels displayed in Fig. 6. Averaged PSNR for 10 realizations of the noise are reported in Table 1, denoised images and detected contours are displayed in Fig. 7 (1st and 3rd rows) for one realization of the noise.

As expected, the PSNR decreases as the noise level increases. Further, large error bars are observed mainly due to the variability of the gradient descent scheme in Algorithm 5, the realizations of the Monte Carlo vector, or the SL-PAM non-convex minimization procedure.

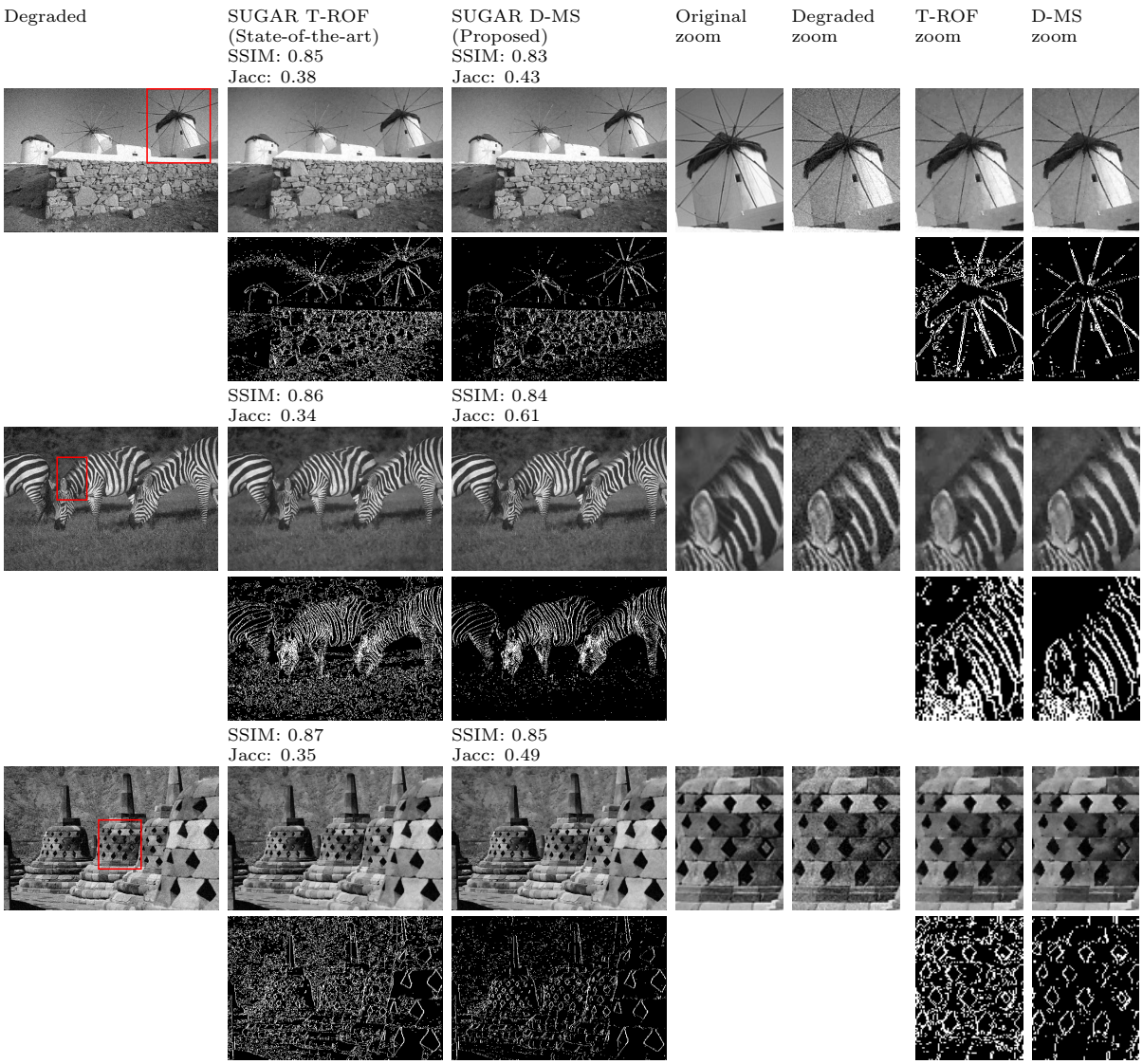


Figure 8: Comparisons between SUGAR T-ROF and SUGAR D-MS for noisy images extracted from the BSD69 dataset [12].

Table 1: PSNR values with 95% confidence interval for true noise level σ and for estimated noise level $\hat{\sigma}$.

σ	Losange		Ellipse	
	True σ	Estimated $\hat{\sigma}$	True σ	Estimated $\hat{\sigma}$
0.01	43.85 ± 0.06	43.64 ± 0.12	41.77 ± 0.06	41.08 ± 0.25
0.03	38.94 ± 0.12	38.89 ± 0.13	31.80 ± 2.73	33.64 ± 2.23
0.05	34.33 ± 0.70	34.71 ± 0.12	26.05 ± 2.92	26.37 ± 2.80
0.07	31.60 ± 0.52	30.55 ± 1.78	26.73 ± 2.46	22.64 ± 2.68
0.1	28.86 ± 0.47	27.24 ± 1.84	21.25 ± 3.03	22.61 ± 3.33

H.3 Impact of the estimation of σ

On real data, the noise level σ needed to implement *Averaged SUGAR D-MS* is unknown. The most usual method to estimate the noise standard deviation is the *median absolute deviation* (MAD) of 2D discrete wavelet coefficients [28]:

$$\hat{\sigma} = \frac{\mathrm{MAD}\left(\left\{|\psi_{H,k}|, |\psi_{V,k}|, |\psi_{D,k}| | k \in \{1, \ldots, \frac{N}{4}\}\right\}\right)}{0.6745}, \tag{53}$$

where $\psi_{H,k}, \psi_{V,k}, \psi_{D,k}$ are the three wavelets coefficents (horizontal, vertical and diagonal) at the finest scale. Table 1 presents the PSNR values with the estimated noise level $\hat{\sigma}$. In addition, some estimates are depicted in Fig. 7 (2nd and 4th rows).

The results with either estimated or true noise level are similar, attesting the robustness of *Averaged SUGAR D-MS* to the estimation of the noise level.

H.4 Real-world images

The proposed automated joint denoising and contour detection procedure *Averaged SUGAR D-MS* is evaluated on real-world images extracted from BSD69 dataset [12] degraded with a Gaussian noise with $\sigma = 0.05$. In our experiments we set $R = 5$ and σ has been estimated from noisy data following (53). Denoised images and contours provided by the proposed data-driven *Averaged SUGAR D-MS* strategy are compared with those yield by SUGAR T-ROF (a two-step procedure, consisting in, first, a piecewise constant denoising with automated tuning of the regularization parameter [3], followed by an iterative thresholding procedure [2]). In Fig. 8, we can observe that the denoising performance are very close for both procedures (in terms of SSIM, SUGAR T-ROF is slightly better) while the contour detection is significantly improved with the SUGAR D-MS procedure (which is confirmed when computing Jaccard index w.r.t contours obtained from the original image).

References

[1] Bertasius G, Shi J, Torresani L (2015) Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4380–4389

[2] Cai X, Steidl G (2013) Multiclass segmentation by iterated ROF thresholding. In: International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer, pp 237–250

[3] Deledalle CA, Vaiter S, Fadili J, et al (2014) Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection 7(4):2448–2487

[4] Foare M, Pustelnik N, Condat L (2019) Semi-linearized proximal alternating minimization for a discrete Mumford–Shah model. IEEE Trans Image Process 29(1):2176–2189

[5] Geman S, Geman D (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans Pattern Anal Match Int (6):721–741

[6] Jaccard P (1901) Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. Bull Soc Vaudoise Sci Nat 37:241–272

[7] Kiefer L, Storath M, Weinmann A (2020) PALMS Image Partitioning-A New Parallel Algorithm for the Piecewise Affine-Linear Mumford-Shah Model. Image Processing On Line 10:124–149

[8] Mumford DB, Shah J (1989) Optimal approximations by piecewise smooth functions and associated variational problems. Commun Pure Appl Math 42(5):577 – 685

[9] Pascal B, Pustelnik N, Abry P (2021) Strongly convex optimization for joint fractal feature estimation and texture segmentation. Appl Comp Harm Analysis 54:303–322

[10] Pascal B, Vaiter S, Pustelnik N, et al (2021) Automated data-driven selection of the hyperparameters for Total-Variation based texture segmentation. J Math Imaging Vis pp 1–30

[11] Ramani S, Blu T, Unser M (2008) Monte-carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms. IEEE Trans Image Process 17(9):1540–1554

[12] Roth S, Black MJ (2009) Fields of experts. International Journal of Computer Vision 82(2):p.205–229

[13] Stein C (1981) Estimation of the mean of a multivariate normal distribution. Ann Stat 9(6):1135–1151

[14] Storath M, Weinmann A, Demaret L (2014) Jump-sparse and sparse recovery using Potts functionals. IEEE Trans Signal Process 62(14):3654–3666

[15] Zach C, Häne C (2017) Discretized convex relaxations for the piecewise smooth Mumford-Shah model. In: Proc. Energy Minimization Methods in Computer Vision and Pattern Recognition, pp 548–563

[16] Ammanouil R, Ferrari CA.and Richard (2018) ADA-PT: An adaptive parameter tuning strategy based on the weighted stein unbiased risk estimator. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp 4449–4453

[17] Ammanouil R, Ferrari A, Mary D, et al (2019) A parallel and automatically tuned algorithm for multispectral image deconvolution. Monthly Notices of the Royal Astronomical Society 490(1):37–49

[18] Attouch H, Bolte J, Redont P, et al (2010) Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality. Mathematics of operations research 35(2):438–457

[19] Babacan SD, Molina R, Katsaggelos AK (2009) Variational Bayesian blind deconvolution using a total variation prior. IEEE Trans Image Process 18(1):12–26

[20] Bauschke HH, Combettes PL (2017) Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, New York

[21] Blake A, Zisserman A (1987) Visual reconstruction. MIT press

[22] Bolte J, Sabach S, Teboulle M (2014) Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Mathematical Programming 146(1):459–494

[23] Chambolle A (1995) Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations. SIAM Journal on Applied Mathematics 55(3):827–863

[24] Chaux C, Duval L, Benazza-Benyahia A, et al (2008) A nonlinear Stein-based estimator for multichannel image denoising. IEEE Trans Signal Process 56(8):3855–3870

[25] Combettes PL, Pesquet JC (2011) Proximal splitting methods in signal processing. In: Fixed-point algorithms for inverse problems in science and engineering. Springer, New York, p 185–212

[26] Curtis FE, Mitchell T, Overton ML (2017) A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles. Optim Methods Softw 32(1):148–181

[27] Dobigeon N, Tourneret JY, Davy M (2007) Joint segmentation of piecewise constant autoregressive processes by using a hierarchical model and a Bayesian sampling approach. IEEE Trans Signal Process 55(4):1251–1263

[28] Donoho DL (1995) Denoising by soft-thresholding. IEEE Trans Inform Theory 41(3):613–627

[29] Donoho DL, Johnstone JM (1994) Ideal spatial adaptation by wavelet shrinkage. Biometrika 81(3):425–455

[30] Eldar YC (2008) Generalized SURE for exponential families: Applications to regularization. IEEE Trans Signal Process 57(2):471–481

[31] Frecon J, Pustelnik N, Dobigeon N, et al (2017) Bayesian Selection for the ℓ_2-Potts Model Regularization Parameter: Constant Signal Denoising. IEEE Trans Signal Process 65(25):5215–5224

[32] Girard A (1989) A fast 'Monte-Carlo cross-validation' procedure for large least squares problems with noisy data. Numerische Mathematik 56(1):1–23

[33] Golub GH, Heath M, Wahba G (1979) Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics 21(2):215–223

[34] Golub GH, Hansen PC, O'Leary DP (1999) Tikhonov regularization and total least squares. SIAM journal on matrix analysis and applications 21(1):185–194

[35] Lucas C, Pascal B, Pustelnik N, et al (2022) Hyperparameter selection for Discrete Mumford-Shah. to be specified

[36] Marin Z, Batchelder KA, Toner BC, et al (2017) Mammographic evidence of microenvironment changes in tumorous breasts. Medical Physics 44(4):1324–1336.

[37] Molina R, Nunez J, Cortijo FJ, et al (2001) Image restoration in astronomy: A Bayesian perspective. IEEE Signal Proc Mag 18(2):11–29

[38] Parikh N, Boyd S (2014) Proximal algorithms. Foundations and Trends® in Optimization 1(3):127–239

[39] Pascal B, Pustelnik N, Abry P, et al (2020) Parameter-free and fast nonlinear piecewise filtering: application to experimental physics. Ann Telecommun 75(11):655–671

[40] Pesquet JC, Benazza-Benyahia A, Chaux C (2009) A SURE approach for digital signal/image deconvolution problems. IEEE Trans Signal Process 57(12):4616–4632

[41] Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena 60(1-4):259–268

[42] Shen X, Ye J (2002) Adaptive model selection. J Am Stat Assoc 97(457):210–221

[43] Stone M (1978) Cross-validation: A review. Statistics: A Journal of Theoretical and Applied Statistics 9(1):127–139

[44] Tikhonov AN, Goncharsky AV, Stepanov VV, et al (2013) Numerical methods for the solution of ill-posed problems, vol 328. Springer Science & Business Media

[45] Vacar C, Giovannelli JF (2019) Unsupervised joint deconvolution and segmentation method for textured images: a Bayesian approach and an advanced sampling algorithm. EURASIP J Adv Signal Process 2019(1):17

[46] Ye J (1998) On measuring and correcting the effects of data mining and model selection. J Am Stat Assoc 93(441):120–131