Adaptive Regularization with the Structure Tensor

V. Estellers, S. Soato, X. Bresson

Abstract

Natural images exhibit geometric structures that are informative of the properties of the underlying scene. Modern image processing algorithms respect such characteristics by employing regularizers that capture the statistics of natural images. For instance, Total Variation (TV) respects the highly kurtotic distribution of the pointwise gradient by allowing for large magnitude outlayers. However, the gradient magnitude alone does not capture the directionality and scale of local structures in natural images. The structure tensor provides a more meaningful description of gradient information as it describes both the size and orientation of the image gradients in a neighborhood of each point. Based on this observation, we propose a variational model for image reconstruction that employs a regularization functional adapted to the local geometry of image by means of its structure tensor. Our method alternates two minimization steps: first, robust estimation of the structure tensor as a semi-definite program; second, reconstruction of the image with an adaptive regularizer defined from this tensor. This two-step procedure allows us to extend anisotropic diffusion [1] into the convex setting and develop robust, efficient, and easy-to-code algorithms for image denoising, deblurring, and compressed sensing. Our method extends naturally to non-local regularization, where it exploits the local self-similarity of natural images to improve non-local TV [2] and diffusion operators [3]. Our experiments show a consistent accuracy improvement over classic regularization.

I. Introduction

We consider the problem of reconstructing an image u from a set of noisy measurements f. To solve this task, we adopt a variational formulation and describe the unknown image as the solution of an optimization problem. Our reconstruction method is thus characterized by a minimization problem and a numerical algorithm to solve it.

The minimization is defined by an objective functional with a data and a prior term:

$$\min_u \ \mathcal{R}(u) + \frac{\alpha}{2}\|Au - f\|_2^2, \tag{1}$$

where the minimization variable, u, is a real-valued function defined over the image domain Ω. The data term couples u with the available measurements f and depends on the problem at hand. The ℓ_2-norm $\|Au - f\|_2^2$ penalizes deviation from the forward linear model $Au = f$ caused by unmodeled phenomena commonly referred as noise. Image denoising, deblurring, and compressed-sensing reconstruction, for instance, follow this model. The prior term $\mathcal{R}(u)$ encodes assumptions about the unknown that are necessary to overcome the ill-posed nature of the problem; it is formalized as a regularization functional that favors solutions exhibiting characteristics of natural images, and its study is one of the cores of this paper.

Total Variation (TV) and Dirichlet functionals are common regularizers in image reconstruction for both modeling and numerical reasons. Both functionals penalize the magnitude of the gradient at each point by capturing the point-wise statistics of natural images: TV models piecewise-constant images with large homogeneous regions separated by a sparse set of edges, while Dirichlet favors homogeneous image variations. As a result, Dirichlet regularization is not able to recover sharp edges, while TV produces piecewise-constant images with staircase artifacts. As optimization problems, nevertheless, both regularizers result in convex functionals that have unique solutions, do not suffer from local minima, and can be efficiently minimized numerically.

In image denoising, TV can be viewed as a regularization technique as well as a diffusion scheme driven by the geometry of the image [6]. This connection appears when we solve the variational problem (2), known as Rudin-Osher-Fatemi (ROF) model [7], with gradient descent.

$$\min_u \int_\Omega |\nabla u| + \frac{\alpha}{2}\|u - f\|_2^2. \tag{2}$$

Euler-Lagrange give us the optimality conditions of (2) in terms of a PDE with the following gradient descent flow:

$$\frac{\partial u}{\partial t} = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) - \alpha(u - f). \tag{3}$$

Minimization of TV gives the term $\operatorname{div}\left(|\nabla u|^{-1}\nabla u\right)$ that smooths the image's level curves with mean curvature flow and corresponds to a non-linear diffusion with coefficient $|\nabla u|^{-1}$. The diffusion is thus adapted to the geometry of the image in terms of the image gradient: it is inhibited close to image edges where $|\nabla u|$ is large, and encouraged in homogeneous areas with small variations. The direction of the diffusion, however, remains constant and parallel to ∇u over the entire image domain, i.e., it is isotropic. This implicit isotropy of TV introduces an additional prior on u, isotropic variations of its level curves, that is not reflected in the statistics of natural images.

To study anisotropic diffusion, we follow Weickert [1] and consider diffusion with a general diffusion tensor T with

$$\frac{\partial u}{\partial t} = \operatorname{div}(T\nabla u), \qquad T(x) \in \mathbb{S}^2_+, \tag{4}$$

where $\mathbb{S}^2_+$ is the symmetric semi-definite positive cone. To encourage diffusion along the edges of the image but not across them, Weickert [1] defines $T(x)$ as a function of the structure tensor of the image at each point.

The structure tensor $J = J(x)$ summarizes the orientation of the image gradient in a neighborhood of a point. To study orientations instead of directions, this tensor first identifies gradients that differ only by their sign with the outer product

$$J_0(x) = \nabla u(x) \otimes \nabla u(x). \tag{5}$$

J_0 encodes directional information of u at the single point and ignores neighborhood information. To describe directions at a larger scale, J is defined as a local average of J_0 by convolution with a Gaussian kernel k_ρ, that is, $J_\rho = k_\rho * J_0$. As a result, the structure tensor $J_\rho(x) \in \mathbb{S}^2_+$ captures geometric neighborhood information and is more informative than the pointwise magnitude of $\nabla u(x)$ used in TV regularization.

The importance of the structure tensor stems from its eigendecomposition, which summarizes the distribution of image gradients in the neighborhood of each point. Let $\boldsymbol{q_1}, \boldsymbol{q_2}$ be the orthonormal eigenvectors of $J_\rho(x)$ and $\lambda_1 \geq \lambda_2 \geq 0$ its eigenvalues, then $\boldsymbol{q_1}$ gives the direction that is maximally aligned with the image gradient in the neighborhood of x, $\boldsymbol{q_2}$ the direction of maximum coherence, and λ_1, λ_2 measure the contrast in these directions. Geometrically, $\{\lambda_i \boldsymbol{q}_i\}$ are the axis of an ellipse fitted to the distribution of image gradients in the neighborhood of a point. We can thus redress this ellipse to a circle by a local change of coordinates in the image domain: $s(x) = \Lambda^{-1/2} Q^T x$ with $Q = [\boldsymbol{q}_1 \boldsymbol{q}_1],\ \Lambda = \operatorname{diag}(\lambda_1, \lambda_2)$. With this transform, $u \circ s$ behaves isotropically around x and fits isotropic regularizers like TV. This suggests a method to design regularizers adapted to the geometry of each image by means of its structure tensor. This is the goal of the paper.

A. Related Work

Anisotropic diffusion first became popular in the 90s with the work of Perona and Malik [8] in image denoising, but it has recently been replaced by convex regularizers amenable to faster minimizations than PDE-evolution techniques.

The use of the structure tensor to denoise images was introduced by Weickert [9], and several extensions have recently followed. Grasmair and Lenzen [10] proposed an anisotropic regularizer based on the structure tensor of the image. Their technique extends only TV for image denoising and relies on a slow minimization algorithm based on gradient descent. With a similar flavor, Roussos and Maragos [11] generalized the Beltrami flow [12] based on the eigendecomposition of the structure tensor. Their focus is on diffusion not regularization, and their denoising technique cannot handle non-differentiable functionals and suffers from slow numerical algorithms. To overcome these limitations, the authors simplified their regularizer in [13] and proposed efficient algorithms for image denoising and deblurring; their new regularizer, however, only depends on the eigenvalues of the struture tensor and does not makes use of the directional information available in the eigenvectors. In contrast, our technique is able to exploit both eigenvector and eigenvalue information by splitting the reconstruction in two: we first estimate the structure tensor of the image and define an adaptive regularizer for it, and we then reconstruct the image with this regularizer. In our technique, these two steps are iterated to improve accuracy.

Previous diffusion or regularization techniques based on the structure tensor [1], [10], [11], [13] are all implicitly designed for applications where the tensor can be reliably estimated by Gaussian smoothing; our technique, on the other hand, investigates reconstructions with fewer measurements where the structure tensor requires robust estimation techniques. For this reason, we formulate the estimation of the structure tensor as a regularization problem in the positive semi-definite cone, generalizing the Gaussian smoothing of [1], [10], [11], [13]. None of the existing models, moreover, has been extended into the non-local setting [2], [3] to improve reconstruction of textured images, as we propose in Section IV.

More distantly related to our work are regularization functionals that eliminate the staircase effect of TV by locally varying the regularization weight [14]–[16] or by penalizing second-order differential operators, like the Total Generalized Variation of [17]. Compared to the former, our regularizer also defines a varying regularization weight, but this is done implicitly by defining the diffusion tensor as a function of the eigenvalues of the structure tensor of the image. Comparison to Total Generalized Variation requires comparison of the image priors, as our model promotes piecewise-constant reconstructions while Total Generalized Variation assumes piecewise-linear images. Such a comparison is an open question that depends on each particular image and application.

B. Contributions

Our contributions are thus fourfold. First, design of an adaptive regularizer that penalizes image variations taking into account the local structure of the image. The proposed regularizer generalizes

anisotropic TV [10], [18] and Dirichlet [1] penalties into a single functional. Compared to anisotropic diffusion [1], [10], [11], the proposed approach allows for faster numerical algorithms based on convex optimization, instead of relying on explicit evolution of PDEs. This allows us to recover non-smooth solutions and solve problems not handled by diffusion techniques, like image reconstruction in compressed sensing. Second, we propose a robust estimate of the structure tensor by regularization of J_0, instead of the local averaging of [1], [10], [11], [13] that ignores image content. To this purpose, the estimation of the structure tensor is re-formulated as a semi-definite program (SDP). Third, we extend our technique to non-local regularization and improve non-local TV and diffusion operators [2], [3]. Finally, we develop efficient algorithms for the estimation the structure tensor and for the reconstruction of images in denoising, deblurring, and compressed-sensing applications.

The rest of the paper is organized as follows: Section II describes the reconstruction of images and their structure tensors, and Section III develops minimization algorithms for the resulting problems. Similarly, Section IV extends our technique to non-local regularization, and Section V develops numerical algorithms for it. Finally, experimental results are presented in Section VI and conclusions drawn in Section VII.

II. Adaptive Image Reconstruction

The main hypothesis underlying our regularization model is that an estimate of the structure tensor of an image can significantly improve its reconstruction. This observation raises two main questions: (i) how to estimate the structure tensor from noisy measurements; (ii) how to incorporate the information of the structure tensor into the reconstruction model (1). The answer that we propose is a two-step iterative method.

In the first step of each iteration, we estimate the structure tensor and adapt the regularizer $\mathcal{R}$ to the particular geometry of the image at each point; the result is the regularizer $\mathcal{R}_J$. In the second step, we use this regularizer in the reconstruction model (1) to obtain an estimate of the image. The process is iterated and we obtain the following two-step iterative method:

$$\begin{cases} J_k = \underset{J(x)\in\mathbb{S}^2_+}{\arg\min}\ \mathcal{G}\left(J\right) + \dfrac{\beta}{2}\|J-\hat{J}\|^2_{F,2} \quad \hat{J}=\nabla u_{k-1}\otimes\nabla u_{k-1} \\ u_k = \underset{u}{\arg\min}\ \mathcal{R}_{J_k}(u) + \dfrac{\alpha}{2}\|Au-f\|_2^2, \end{cases}$$

where α, β are model parameters, $\mathcal{G}\left(J\right)$ a regularizer on the structure tensor, and $\|J-\hat{J}\|^2_{F,2} = \int_\Omega \|J(x)-\hat{J}(x)\|^2_F$ measures the distance between the structure tensor and a coarse estimate $\hat{J}$ in the image domain Ω with the Frobenius norm $\|\cdot\|_F$.

Combining the two stages into one leads to a reconstruction model difficult to analyze in theory and minimize in practice. The main difficulty lies in the dependence of $\mathcal{R}_J$ on u through convolution of image derivatives and pixel-wise matrix inversion, which results in a non-convex regularizer. As a result, the minimization of the single-step method is not convex and suffers from local minima and slow-convergence algorithms. As an alternative, we estimate the structure tensor from a coarse image estimate and refine it through regularization. We then define the adaptive regularizer and reconstruct a better image estimate. The novelty of our approach lies in decomposing the problem into a sequence of convex minimizations each of which can be efficiently solved by adapting standard primal-dual algorithms to the characteristics of the problem. This *divide and conquer* strategy currently lacks a proof of convergence, but our promising experimental results motivate future analysis.

A. Adaptive Anisotropic Regularization

TV and Dirichlet regularizers quantify the variations of u with a scalar function of the magnitude of ∇u. We summarize both functionals with the Huber penalty g_γ[1] [19]:

$$\mathcal{R}(u) = \int_\Omega g_\gamma(\nabla u), \quad g_\gamma(\nabla u) = \begin{cases} \frac{|\nabla u|^2}{2\gamma} & \text{if } |\nabla u| < \gamma \\ |\nabla u| - \frac{\gamma}{2} & \text{if } |\nabla u| \geq \gamma \end{cases}, \tag{6}$$

The regularizer $\mathcal{R}$ is convex in u and behaves like the Dirichlet penalty in homogeneous regions, where $|\nabla u| < \gamma$, while it leads to TV regularization close to the image edges, where $|\nabla u| \geq \gamma$. As a result, we can obtain Dirichlet regularization by setting $\gamma > \||\nabla u|\|_\infty$, and TV with $\gamma = 0$. In-between, a small γ recovers sharp edges because it does not over-penalize high variations of u, while it avoids the staircase effect of TV because it removes its singularity at $\nabla u = 0$.

To find the link with diffusion PDEs, we use Euler-Lagrange to derive the descent flow that minimizes $\mathcal{R}(u)$; that is,

$$\frac{\partial u}{\partial t} = \operatorname{div}\left(\frac{1}{\max(\gamma, |\nabla u|)} \nabla u\right). \tag{7}$$

The diffusion is still parallel to ∇u and isotropic. In fact, to define a regularizer leading to anisotropic diffusion (4) we need to introduce a diffusion tensor in $\mathcal{R}$.

Given a diffusion tensor $\tilde{T} \in \mathbb{S}^2_+$, we consider the regularizer

$$\mathcal{R}_T(u) = \int_\Omega g_\gamma(S \nabla u), \qquad S(x) = \sqrt{\tilde{T}(x)} \tag{8}$$

[1]In an abuse of notation, we use the same expression for the Huber norm $g_\mu(\boldsymbol{z})$ throughout the paper, even though the dot product operator in $|\boldsymbol{z}|^2 = \sqrt{\boldsymbol{z} \cdot \boldsymbol{z}}$ is defined by the Hilbert space associated with variable $\boldsymbol{z}$.

Its descent flow leads to an anisotropic diffusion parallel to $\tilde{T}$:

$$\frac{\partial u}{\partial t} = \operatorname{div}\left(T(x)\nabla u\right), \qquad T(x) = \frac{1}{\max(\gamma, |S\nabla u|)}\tilde{T}(x), \tag{9}$$

We now adapt this regularizer to each image by defining T as a function of the structure tensor of the image; this defines our adaptive regularizer $\mathcal{R}_J = \mathcal{R}_{T(J)}$.

Let $J(x)$ be the estimated structure tensor of the image at point x, with eigenvectors $Q(x) = [\boldsymbol{q}_1(x)\ \boldsymbol{q}_2(x)]$ and diagonal matrix of eigenvalues $\Lambda(x)$; we define the diffusion tensor $T(x) = Q(x)\Gamma(x)Q^T(x)$ with the same orientation as $J(x)$ and positive eigenvalues $\Gamma(x) = \Gamma(\Lambda(x))$. To justify our choice of Γ, we make use of the geometric interpretation of $\{\lambda_i(x)\boldsymbol{q_i}(x)\}$ as the axes of the ellipse that describes the orientation and size of the image gradient locally. For a correct reconstruction, our adaptive regularizer first redresses this ellipse to the circle before measuring the image smoothness with the isotropic regularizer $\mathcal{R}$; this is accomplished with $\Gamma(x) = \Lambda^{-1}(x)$, which corresponds to a change of coordinates, $S(x) = \Lambda^{-\frac{1}{2}}(x)Q^T(x)$.

Finally, we need to define the tensor $\tilde{T} = \max(\gamma, |S\nabla u|)T$. We observe that $\tilde{T}$ is a scaled version of T, with a spatially variable factor that depends on the unknown image. We can still approximate its spatial variability from

$$|S\nabla u| = \sqrt{(\nabla u)^T T \nabla u} = \sqrt{\lambda_1^{-1}(\boldsymbol{q_1}\cdot\nabla u)^2 + \lambda_2^{-1}(\boldsymbol{q_2}\cdot\nabla u)^2},$$

where $(\boldsymbol{q_i}\cdot\nabla u)^2 \propto |\lambda_i|^2$ for the real –not the estimated– structure tensor of u. We assume that an approximate proportionality holds for our estimate, $|S\nabla u| \propto \sqrt{\lambda_1+\lambda_2}$, and we let model parameter α absorb the remaining constant scaling.

In brief, given an estimate of the structure tensor, we compute its eigen-decomposition $J = Q\Lambda Q^T$ at each point and define our adaptive regularizer

$$\mathcal{R}_J(u) = \int_\Omega g_\gamma(S\nabla u), \quad S = \max(\sqrt{\gamma}, \sqrt[4]{\lambda_1+\lambda_2})\ \Lambda^{-\frac{1}{2}}Q^T.$$

B. Robust Estimation of the Structure Tensor

Each iteration the structure tensor is estimated in two steps. First, we obtain a noisy point-wise estimate $\hat{J}$ from the image estimate u_{k-1}, and we then regularize it to obtain J_k.

The first estimate of the structure tensor is computed directly as the outer product of image gradients as in Equation (5), i.e.,

$$\hat{J} = \nabla u_{k-1} \otimes \nabla u_{k-1}. \tag{10}$$

The definition of $\hat{J}$ in terms of derivatives makes it very sensitive to noise and artifacts in u_{k-1} and calls for regularization. Compared to the Gaussian smoothing of previous techniques [1], [10], [11], [13], our regularization technique is adapted to the geometry of $\hat{J}$ and the positive semi-definite cone.

As in any regularization model, we have a smoothness and a data term: the smoothness term is a generalization of TV and Dirichlet penalties, while the data term penalizes deviations from $\hat{J}$ in the positive semi-definite cone.

Let ∇J be the Jacobian matrix of component-wise gradients and $|\cdot|_F$ the Frobenius norm, our smoothness term

$$\mathcal{G}(J) = \int_D g_\mu(\nabla J), \qquad g_\mu(\boldsymbol{z}) = \begin{cases} \frac{|\boldsymbol{z}|_F^2}{2\mu} & \text{if } |\boldsymbol{z}|_F < \mu \\ |\boldsymbol{z}|_F - \frac{\mu}{2} & \text{if } |\boldsymbol{z}|_F \geq \mu \end{cases} \tag{11}$$

uses the Huber norm g_μ to allow sharp discontinuities at the boundaries of textured regions and small variations in within. In fact, setting $\mu = 0$ leads to vectorial TV [20], while $\mu > \||\nabla \hat{J}|_F\|_\infty$ provides a Dirichlet regularization equivalent to Gaussian smoothing [1], [10], [11], [13]. The proposed $\mathcal{G}$ is also convex in J, which simplifies the minimization.

We propose two models to regularize $\hat{J}$ in the space of positive semi-definite matrices. The first model introduces the constraint $J(x) \in \mathbb{S}^2_+$ in the minimization and handles it through pixel-wise projection into $\mathbb{S}^2_+$, that is,

$$J_k = \arg\min_J \ \mathcal{G}(J) + \frac{\beta}{2}\|J - \hat{J}\|^2_{F,2} \ \text{ s.t. } \ J(x) \in \mathbb{S}^2_+. \tag{12}$$

The second model takes into account that also $\hat{J}(x) \in \mathbb{S}^2_+$, and adapts the ℓ_2 penalty in (12) to the metric of $\mathbb{S}^2_+$. To this purpose, we use the Lie-group structure of $\mathbb{S}^2_{++}$ to define a data term that measures the geodesic distance in $\mathbb{S}^2_{++}$. This kind of regularization was originally proposed for diffusion tensors in MRI [21], [22], but the resulting models are computationally too expensive for large images. To overcome this issue, [23] develops log-Euclidean metrics that approximates the metric of $\mathbb{S}^2_{++}$ in the matrix-logarithm domain. Log-Euclidean metrics are based on the observation that the logarithm transforms elements of $\mathbb{S}^2_{++}$ into symmetric matrices in $\mathbb{S}^2$, matrix exponentiation maps them back to $\mathbb{S}^2_{++}$ in a one-to-one mapping, and relative distance relations –geodesic in $\mathbb{S}^2_{++}$ and Euclidean in $\mathbb{S}^2$– are preserved by this transform[2]. Consequently, regularization can be carried out in the logarithm domain

[2]This method introduces a bias towards positive definite matrices, instead of positive semi-definite ones; we handle this limitation by setting the null eigenvalues of $\hat{J}$ to $\epsilon = 10^{-6}$ in the logarithmic transform and thresholding them in the back-projection.

with Euclidean metrics, as follows:

$$\log J_k = \underset{\log J}{\arg\min}\ \mathcal{G}\left(\log J\right) + \frac{\beta}{2}\|\log J - \log \hat{J}\|_{F,2}^2. \tag{13}$$

Defining the minimization variable to be $\log J$, we solve (13) as a standard vectorial regularization and recover the structure tensor by matrix exponentiation $J_k(x) = \exp(\log J_k(x))$. For details on log-Euclidean metrics and other approaches to fast SDP matrix regularization, we refer the reader to [23], [24].

C. Enhanced Image Reconstruction

Given an estimate of the structure tensor, J_k, we reconstruct an image by solving the following minimization problem:

$$u_k = \underset{u}{\arg\min}\ \mathcal{R}_{J_k}(u) + \frac{\alpha}{2}\|Au - f\|_2^2. \tag{14}$$

In image denoising, A is the identity operator, in deblurring A is the blurring kernel, and in compressive sensing A is the sampling matrix.

In the case of image denoising, several methods emerge as special cases of the proposed model: standard diffusion corresponds to $\beta = 0,\ \gamma > \||\nabla u^0|\|_\infty$, TV corresponds to $\beta = 0,\ \gamma = 0$, and the anisotropic diffusion of Weickert [9] is obtained with $\beta > 0, \mu > \||\nabla \hat{J}|\|_\infty, \gamma = 0$.

III. Numerical Minimization

To efficiently solve the minimization problems (12) and (14), we re-formulate them as constraint minimizations with independent variables for each term, form the associated Lagrangians, and solve the resulting saddle-point problems with a primal-dual algorithm. To simplify notation, we remove the k subindexes in u_k and J_k that indicate the iterations.

The minimizations (12) and (14) involve a quadratic term and a regularizer similar to the ROF model [7]. Consequently, our algorithms apply a strategy to overcome the dependency on derivatives and the non-linearity in $\mathcal{R}_J$ similar to the multitude of algorithms for this model. The first algorithms for the ROF model [7] solved the minimization by explicit evolution of its descent-flow PDE, this lead to slow algorithms with time steps limited by Courant-Friedrichs-Lewy stability conditions [25]. To overcome this limitation, techniques based on variable splitting and constrained optimization have been recently introduced.

A popular class of methods is based on the dual formulations of ROF, like Chambolle's projection method [26], or the primal-dual approaches [27]–[29]. Other options are based on variable-splitting and

constrained optimization; which is solved by quadratic-penalties [30], Bregman iterations [31], [32], or augmented Lagrangians [33]. Among them, we choose the primal-dual approach of [29] because it leads to minimization problems with closed-form solutions computationally cheaper to compute than the splitting methods [30], [31], [33] and it can be easily accelerated by adapting its time steps σ, τ to guarantee an optimal rate of convergence.

A. Discretization and Notation

In the numerical minimization, we discretize Ω with a regular grid of size $n = n_x \times n_y$ and describe functions in Ω by the values taken at each sample point, i.e., column vectors in $\mathbb{R}^n$. Vector or matrix fields with k components are rasterized in the same manner into vectors in $\mathbb{R}^{nk}$. We use forward differences to compute the discrete gradients and backward differences for the divergence in order to preserve the adjoint relationship $\mathrm{div} = -\nabla^*$. With this notation, we can efficiently compute the spatial derivatives multiplying the discrete functions with the sparse finite difference matrices. Finally, $|\cdot|$ denotes the magnitude of a vector field defined in the image domain Ω, while $\|\cdot\|_2$ is the ℓ_2 norm in its Hilbert space. The dot product of vector fields is denoted by $\boldsymbol{v_1} \cdot \boldsymbol{v_2}$, while its inner product is $\langle u_1, u_2 \rangle$.

B. Reconstruction with Adaptive Regularizer

To reconstruct the image we need to solve the problem:

$$\min_u \ \frac{\alpha}{2}\|Au - f\|_2^2 + \int_\Omega g_\gamma(S\ \nabla u). \tag{15}$$

Our algorithm exploits the convexity of each of the terms in (15) to re-formulate it as a saddle-point problem that is easy to solve in the primal and dual variables independently.

To find the saddle-point formulation, we re-write (15) as the following constrained minimization:

$$\min_u \ \frac{\alpha}{2}\|Au - f\|_2^2 + F(\boldsymbol{d}) \ \ \text{s.t.} \ \ \boldsymbol{d} = Ku \tag{16}$$

where K is the linear operator defined by the combination of the spatial gradient with the pixel-dependent linear transform S, i.e., $Ku(x) = S(x)\nabla u(x)$, and $F(\boldsymbol{d}) = \int_\Omega g_\gamma(\boldsymbol{d})$.

The Lagrangian of (16) is obtained by introducing a dual variable $\boldsymbol{y}$ for the constraint $\boldsymbol{d} = Ku$; it reads

$$\mathcal{L}(\boldsymbol{y}) = \min_{u,\ d} \ \frac{\alpha}{2}\|Au - f\|_2^2 + F(\boldsymbol{d}) + \langle \boldsymbol{y}, Ku - \boldsymbol{d} \rangle.$$

The Lagrangian can be simplified in terms of the conjugate function $F^\star(\boldsymbol{y}) = \sup_{\boldsymbol{d}} \langle \boldsymbol{y}, \boldsymbol{d}\rangle - F(\boldsymbol{d})$, as follows:

$$\mathcal{L}(\boldsymbol{y}) = \min_u \frac{\alpha}{2}\|Au - f\|_2^2 - F^*(\boldsymbol{y}) + \langle \boldsymbol{y}, Ku\rangle.$$

Finally, convex analysis tells us that the minimization problem (16) is equivalent to the saddle-point problem

$$\min_u \max_{\boldsymbol{y}} \frac{\alpha}{2}\|Au - f\|_2^2 - F^*(\boldsymbol{y}) + \langle \boldsymbol{y}, Ku\rangle. \tag{17}$$

This formulation allows us to apply the primal-dual algorithm of Chambolle-Pock [29], which reduces to solving a sequence of minimization problems in u and $\boldsymbol{y}$ independently. Applied to our problem, these read:

$$\begin{aligned}
\boldsymbol{y}^{t+1} &= \arg\min_{\boldsymbol{y}} \ \sigma F^*(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{y}^t - \sigma K z^t\|_2^2 \\
u^{t+1} &= \arg\min_u \ \tau\alpha\|Au - f\|_2^2 + \|u - u^t + \tau K^*\boldsymbol{y}^{t+1}\|_2^2 \\
z^{t+1} &= \ u^{t+1} + \theta(u^{t+1} - u^n).
\end{aligned}$$

As the problem is uniformly convex, this algorithm can be further accelerated by updating the parameters σ, τ (Alg. 2 [29]). For simplicity, we omit the acceleration updates here.

The efficiency of the proposed algorithm comes from the ability to find closed-form solutions for each of the sub-minimization problems. The derivation of these solutions is detailed next and the procedure summarized in Algorithm 1.

Initialize $u, \boldsymbol{y}, z$ to 0.

Choose $\tau, \sigma > 0$, $\theta \in [0, 1]$.

while $\|u^{t+1} - u^t\| > 1^{-4}$ **do**

$\boldsymbol{y}^{t+1}(i) = \frac{\hat{\boldsymbol{y}}(i)}{max(1+\sigma\gamma, |\hat{\boldsymbol{y}}(i)|)}$, $\quad \hat{\boldsymbol{y}} = \boldsymbol{y}^t + \sigma K z^t$

$u^{t+1} = \frac{\hat{u}+\alpha f}{1+\tau\alpha}$, $\quad \hat{u} = u^t - \tau K^*\boldsymbol{y}^{t+1}$

$z^{t+1} = \ u^{t+1} + \theta(u^{t+1} - u^t)$

Algorithm 1: *Efficient minimization associated to reconstruction model* (15) *for image denoising. Deblurring and compressed sensing reconstructions update* u *with* (21).

If we define $\hat{u} = u^t - \tau K^*\boldsymbol{y}^{t+1}$, the minimization in u is discretized as the least-squares problem

$$\min_u \frac{\tau\alpha}{2}\|Au - f\|_2^2 + \frac{1}{2}\|u - \hat{u}\|_2^2. \tag{18}$$

Its optimality conditions lead to the following linear system, with positive definite matrix:

$$(\tau\alpha A^T A + I_n)u = b \quad \text{with } b = \hat{u} + \alpha A^T f. \tag{19}$$

While in image denoising $A = I_n$ and the solution $u = \frac{\hat{u}+\alpha f}{1+\tau\alpha}$ does not require any matrix inversion, in deblurring and compressed sensing, we need efficient ways to invert the matrix $\tau\alpha A^T A + I_n$.

In image deblurring, Au denotes the convolution of the image u with a Gaussian kernel k. This operation can be computed efficiently in the frequency domain by point-wise multiplication of their Fourier transforms, and the result recovered by the inverse Fourier transform $\mathcal{F}^{-1}$, that is: $Au = \mathcal{F}^{-1}\hat{V}$, with $\hat{V}(i) = \hat{K}(i)\hat{U}(i)$ and $\hat{K},\ \hat{U}$ the 2-dimensional discrete Fourier transforms of k and u. In compressive sensing, the measurements f are random samples in the Fourier domain, and the matrix $A = R\mathcal{F}$ decomposes into a binary mask R and the discrete Fourier transform $\mathcal{F}$.

In both cases, $(\tau\alpha A^T A + I_n)u$ can be written as an entry-wise multiplication in the Fourier domain. As a result, the linear system (19) is re-written in the Fourier domain as

$$\hat{D}_\star(i)\hat{U}(i) = \mathcal{F}[b](i), \quad i = 1, \dots, n, \tag{20}$$

where $\hat{D}_\star$ comes from the Fourier decomposition of $(\tau\alpha A^T A + I_n)$: $\hat{D}_\star(i) = \tau\alpha\hat{K}^2(i) + 1$ for deblurring, and $\hat{D}_\star(i) = \tau\alpha R^2(i) + 1$ for compressed sensing. With the inverse Fourier transform we then solve (20) as $u = \mathcal{F}^{-1}\hat{U}$, with

$$\hat{U}(i) = \frac{\mathcal{F}[b](i)}{\hat{D}_\star(i)}, \qquad\qquad b = \hat{u} + \alpha A^T f. \tag{21}$$

In terms of implementation, the FFT gives a fast and exact $\mathcal{O}(n \log n)$ algorithm to perform discrete Fourier transforms.

In order to solve the minimization in $\boldsymbol{y}$, we use Moreau's identity to rewrite the minimization in terms of F. This gives

$$\boldsymbol{y}^\star = \arg\min_{\boldsymbol{y}} \ \sigma F^*(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 \iff \boldsymbol{y}^\star = \hat{\boldsymbol{y}} - \sigma\boldsymbol{d}^\star$$

with

$$\boldsymbol{d}^\star = \arg\min_{\boldsymbol{d}} \ F(\boldsymbol{d}) + \frac{\sigma}{2}\|\boldsymbol{d} - \frac{\hat{\boldsymbol{y}}}{\sigma}\|_2^2. \tag{22}$$

We can thus simply solve the minimization (22), which is decoupled for each pixel. Its discretization

$$\min_{\boldsymbol{d}(i)} \ \sum_i \ g_\gamma(\boldsymbol{d}(i)) + \frac{\sigma}{2}\left|\boldsymbol{d}(i) - \frac{\hat{\boldsymbol{y}}(i)}{\sigma}\right|^2, \tag{23}$$

requires only the minimization of $h(\boldsymbol{v}) = g_\gamma(\boldsymbol{v}) + \frac{\sigma}{2}|\boldsymbol{v} - \frac{\hat{\boldsymbol{y}}(i)}{\sigma}|^2$ in $\boldsymbol{v} \in \mathbb{R}^2$ for each pixel in the image.

For $\gamma > 0$, the Huber penalty is differentiable and we can compute its derivative with respect to each component of $\boldsymbol{v}$, that is, $\frac{\partial g_\gamma}{\partial \boldsymbol{v}} = \frac{\boldsymbol{v}}{\max(\gamma, \|\boldsymbol{v}\|)}$. Consequently, the optimality conditions for (23),

$$\frac{\boldsymbol{d}(i)}{\max(\gamma, |\boldsymbol{d}(i)|)} + \sigma \boldsymbol{d}(i) - \hat{\boldsymbol{y}}(i) = 0, \tag{24}$$

are solved by

$$\boldsymbol{d}(i) = \frac{\hat{\boldsymbol{y}}(i)}{\sigma} - \frac{1}{\sigma} \frac{\hat{\boldsymbol{y}}(i)}{\max(|\hat{\boldsymbol{y}}(i)|, 1 + \sigma\gamma)}. \tag{25}$$

We then update the dual variable, as follows:

$$\boldsymbol{y}(i) = \frac{\hat{\boldsymbol{y}}(i)}{\max(|\hat{\boldsymbol{y}}(i)|, 1 + \sigma\gamma)}. \tag{26}$$

For $\gamma = 0$, we have $g_\gamma(\boldsymbol{v}) = |\boldsymbol{v}|$, and the minimization corresponds to a well-known non-differentiable problem that is solved by shrinkage; this gives

$$\boldsymbol{d}(i) = \frac{1}{\sigma} \max\Big(|\hat{\boldsymbol{y}}(i)| - 1, 0\Big) \frac{\hat{\boldsymbol{y}}(i)}{|\hat{\boldsymbol{y}}(i)|} \rightarrow \boldsymbol{y}(i) = \frac{\hat{\boldsymbol{y}}(i)}{\max(1, |\hat{\boldsymbol{y}}(i)|)}.$$

In terms of the dual variable $\boldsymbol{y}$, its update rule can be incorporated into the expression (26) with $\gamma = 0$. Putting together the different updates we get Algorithm 1.

C. Regularization of the Structure Tensor

Section II-B presents two models to regularize the estimated structure tensor $\hat{J}$. We detail here the algorithm for model (12), as it can handle model (13) by omitting the projection onto $\mathbb{S}^2_+$.

Model (12) has to solve

$$\min_J \int_\Omega g_\mu(\nabla J) + \frac{\beta}{2} \|J - \hat{J}\|^2_{F,2} \quad \text{s.t.} \quad J(x) \in \mathbb{S}^2_+, \tag{27}$$

We use the same strategy to solve this minimization: we reformulate it as a constrained optimization and solve the associated saddle-point problem with algorithm [29] finding closed-form solutions for each sub-minimization.

The equivalent constrained minimization is

$$\min_J \int_\Omega g_\mu(M) + \frac{\beta}{2} \|J - \hat{J}\|^2_{F,2} \quad \text{s.t.} \quad M = \nabla J, \; J(x) \in \mathbb{S}^2_+.$$

If $G(M) = \int_\Omega g_\mu(M)$, its associated saddle-point problem is

$$\min_{J(x) \in \mathbb{S}^2_+} \max_Y \frac{\beta}{2} \|J - \hat{J}\|^2_{F,2} - G^*(Y) + \langle Y, \nabla J \rangle. \tag{28}$$

The primal-dual algorithm [29] requires the following updates

$$\begin{aligned} Y^{t+1} &= \arg\min_{Y} \ \sigma G^*(Y) + \frac{1}{2}\|Y - Y^t - \sigma\nabla Z^t\|^2_{F,2} \\ J^{t+1} &= \arg\min_{J(x)\in\mathbb{S}^2_+} \frac{\tau\beta}{2}\|J - \hat{J}\|^2_{F,2} + \frac{1}{2}\|J - J^t + \tau\nabla^* Y^{t+1}\|^2_{F,2} \\ Z^{t+1} &= \ J^{t+1} + \theta(J^{t+1} - J^t). \end{aligned}$$

The minimization in J can be written as follows:

$$\min_{J(x)\in\mathbb{S}^2_+} \ \frac{\tau\beta+1}{2}\|J - \tilde{J}\|^2_{F,2}, \quad \tilde{J} = \frac{\tau\beta\hat{J} - J^t + \tau\nabla^* Y^{t+1}}{\tau\beta+1}.$$

The problem is decoupled for each pixel, and it is solved by pixel-wise projecting $\tilde{J}$ onto the semi-definite positive cone. This projection is accomplished by computing the eigenvector decomposition of the matrix $\tilde{J}=\tilde{Q}\tilde{\Lambda}\tilde{Q}^T$ and setting negative eigenvalues to zeros, that is, $J=\tilde{Q}(\tilde{\Lambda})_+\tilde{Q}^T$. As $\tilde{J}$ is a 2×2 symmetric matrix, we have closed-form expressions for its eigendecomposition with computational cost of $\mathcal{O}(n)$.

We use again Moreau's identity for the minimization in Y

$$Y^\star = \arg\min_{Y} \ \sigma G^*(Y) + \frac{1}{2}\|Y - \hat{Y}\|^2_{F,2} \iff Y^\star = \hat{Y} - \sigma M^\star$$

with

$$M^\star = \arg\min_{M} \ G(M) + \frac{\sigma}{2}\|M - \frac{\hat{Y}}{\sigma}\|^2_{F,2}. \tag{29}$$

The discretization of (29) is again decoupled for each pixel as

$$\min_{M(i)} \ \sum_i g_\gamma(M(i)) + \frac{\sigma}{2}\left\|M(i) - \frac{\hat{Y}(i)}{\sigma}\right\|^2_F. \tag{30}$$

It is therefore solved by pixel-wise minimization of the previous function $h(\boldsymbol{v}) = g_\mu(\boldsymbol{v}) + \frac{\sigma}{2}\|\boldsymbol{v} - \frac{\hat{Y}(i)}{\sigma}\|^2_F$, with $\boldsymbol{v} \in \mathbb{R}^{2\times 4}$. The previous solutions are independent of the dimension of $\boldsymbol{v}$ and the parameter μ, and we obtain the same closed-form updates. The final procedure is summarized in Algorithm 2.

IV. Extension to Non-Local Operators

Non-local (NL) regularization has been recently introduced in image processing to exploit the local self-similarity of natural images to improve image reconstruction [2], [34], [35]. To this purpose, NL techniques take into account multiple instances of the same image structure and define the concept of non-local derivatives. Research, however, is still limited to NL equivalents of isotropic TV [2] and Dirichlet regularizers [34], and non-local equivalents to the structure tensor have not been investigated yet. This is the goal of this section.

Initialize $J^0 = \hat{J},\ Y^0 = \mathbf{0},\ Z^0 = J^0$.

Choose $\tau, \sigma > 0,\ \theta \in [0, 1]$.

while $\|J^{t+1} - J^t\| > 1^{-4}$ **do**

$$Y^{t+1}(i) = \frac{\hat{Y}(i)}{\max(\|\hat{Y}(i)\|, 1+\sigma\mu)}, \quad \hat{Y} = Y^t - \sigma \nabla Z^t$$

$$J^{t+1} = \tilde{Q}(\tilde{\Lambda})_+ \tilde{Q}^T,\ [\tilde{Q}, \tilde{\Lambda}] = \mathrm{eig}(\tfrac{\tau\beta\hat{J} - J^t + \tau\nabla^* Y^{t+1}}{\tau\beta+1})$$

$$Z^{t+1} = \ J^{t+1} + \theta(J^{t+1} - J^t)$$

Algorithm 2: *Efficient minimization associated to the estimation of the structure tensor* (27). $J(i),\ Y(i)$ *are the values of the tensor fields* J *and* Y *at pixel* i.

In the study of this question, we consider the recovery model

$$\min_u \ R_G(u) + \frac{\alpha}{2}\|Au - f\|_2^2, \tag{31}$$

where $R_G(u)$ is a NL regularizer, like the NL-TV [2] or the NL-diffusion [34]. Analogously to the local case, we adapt R_G to each image by means of its NL structure tensor.

A. Non-Local Regularization

Non-local TV and Dirichlet regularizers are variational extensions of the NL-means filter [36] for image denoising. They come in both discrete and continuous flavors: in the continuous setting, Gilboa and Osher [2] use graph theory to extended TV to a non-local functional, while Zhou and Schölkopf [3] and Elmoataz et al. [37] use graph Laplacians to define discrete NL operators. These operators have been used in denoising [2], segmentation [38], [39], inpainting [35], deconvolution, and compressive sensing [40], [41], but research is still open to new models, not only to new applications.

Our NL method discretizes the continuous formulation of [2] and considers the image as a graph $G = (\Omega, E)$, where Ω is the set of nodes, one for each pixel in the image, and E is the set of edges connecting the nodes. The edge connecting nodes i and j is weighted by $w(i, j)$, which measures the distance between the two nodes in the metric of the graph. Consequently, two pixels spatially far away in the image domain can be neighbors in the graph and interact if $w(i, j) > 0$, which we write $i \sim j$.

Given an image u, the non-local gradient $\nabla_G u$ at node i is defined as the vector of all directional derivatives associated with the neighbors of i, that is

$$\nabla_G u\ (i, j) = (u(j) - u(i))\sqrt{w(i, j)} \ \ \forall\ j \sim i. \tag{32}$$

$\nabla_G u$ is thus a vector field in $\Omega \times \Omega$ with components indexed by the neighbors of each node. We are interested in sparse graphs where each node is connected to a small number of neighbors, and we can summarize the gradient $\nabla_G u$ at node i as a vector with ν_i effective components – the rest being null.

We can naturally define a dot product of vector fields by simple analogy to the local case,

$$(\boldsymbol{d} \cdot \boldsymbol{e})_G(i) = \sum_{j \sim i} d(i,j) e(i,j), \tag{33}$$

Similarly, the inner product of functions and vector fields is

$$\langle u, v \rangle_G = \sum_{i \in \Omega} u(i) v(i) \qquad\qquad \langle \boldsymbol{d}, \boldsymbol{e} \rangle_G = \sum_i (\boldsymbol{d} \cdot \boldsymbol{e})_G(i).$$

With the previous definitions, the magnitude of a vector field at node i is given by

$$|\boldsymbol{d}|_G(i) = \sqrt{(\boldsymbol{d} \cdot \boldsymbol{d})_G(i)}, \tag{34}$$

Consequently, the NL-TV and Dirichlet functionals of [2], [42] can be written in terms of the Huber penalty, as follows:

$$\mathcal{R}_G(u) = \int_\Omega g_\delta(\nabla_G u), \quad g_\delta(\boldsymbol{z}) = \begin{cases} \frac{|\boldsymbol{z}|_G^2}{2\delta} & \text{if } \ |\boldsymbol{z}|_G < \delta \\ |\boldsymbol{z}|_G - \frac{\delta}{2} & \text{if } \ |\boldsymbol{z}|_G \geq \delta \end{cases}.$$

With these definitions, if we consider only the immediate pixels as neighbors with unit weights, $\mathcal{R}_G$ reduces to the finite-difference discretizations of $\mathcal{R}$. By considering more general neighborhoods, ∇_G incorporates global information and the regularization is improved. For this reason, we follow [36] and define the weighting function by measuring the difference between 5×5 patches around each node, that is,

$$w(i,j) = \exp^{-\frac{\|\mathcal{P}_0(i) - \mathcal{P}_0(j)\|_2^2}{2h^2}}, \tag{35}$$

where h is a scaling factor, and $\mathcal{P}_0(i)$ is an image patch of u_0 centered at pixel i. This weighting function is designed to reduce Gaussian noise while preserving image textures.

With these weights, ∇_G is already adapted to the structure of the image and reconstruction with $\mathcal{R}_G$ exploits similarities between patches. Our NL regularizer goes a step further and exploits the similarity of the image within the patch neighbors.

B. Non-Local Adaptive Image Reconstruction

Mimicking Section II, we can derive a two-step NL reconstruction method. To compute the graph weights, the method is initialized with the reconstruction model (1).

In the first step of each iteration, we estimate the NL structure tensor of the image, J_G, and define an adaptive regularizer $\mathcal{R}_{J_G}$. In the second step, we reconstruct an image with the proposed regularizer $\mathcal{R}_{J_G}$ by solving a convex minimization problem. Iterating the process, we obtain our NL method:

$$\begin{cases} J_{G_k} \leftarrow \text{estimate NL structure tensor from } u_{k-1} \\ u_k = \arg\min_u \ \mathcal{R}_{J_G}(u) + \dfrac{\alpha}{2}\|Au - f\|_2^2 \end{cases}.$$

We initialize u_0 to the solution of its local model for the weight computation (35). The next sections detail each of these steps.

C. Non-Local Adaptive Regularization

By analogy with the local case, we consider a NL anisotropic regularizer defined by a diffusion tensor $\tilde{T}_G$:

$$\mathcal{R}_{J_G}(u) = \sum_{i\in\Omega} g_\delta(S_G \nabla_G u), \qquad S_G(i) = \sqrt{\tilde{T}_G(i)}, \tag{36}$$

where $\tilde{T}_G(i)$ can be effectively reduced to a tensor in $\mathbb{S}_+^{\nu_i}$ by only considering the components of $\nabla_G u$ associated to the neighbors of node i, that is, $\nabla_G u(i) = \nabla_G u(i,\cdot) \in \mathbb{R}^{\nu_i}$. In what follows, we consider this compact notation.

The diffusion PDE of (36) comes from its formulation in the continuous framework [2], where it is easy to show that its descent flow leads to the anisotropic diffusion

$$\frac{\partial u}{\partial t} = \mathrm{div}_G\left(T_G \nabla_G u\right), \quad T_G \ = \ \frac{1}{\max(\delta, |S_G \nabla_G u|)} \tilde{T}_G.$$

The diffusion tensor T_G can again be defined as a function of the NL structure tensor of the image.

Let $J_G(i) \in \mathbb{S}_+^{\nu_i}$ be the estimated structure tensor of the image at node i, and $J_G(i) = Q_G(i)\Lambda_G(i)Q_G^T(i)$ its eigen-decomposition. As in the local case, we define the diffusion tensor with the same orientation as $J_G(i)$ and positive eigenvalues $\Lambda_G^{-1}(i)$. This tensor can then be decomposed as $T_G(i) = S_G^T(i)S_G(i)$, with $S_G(i) = \Lambda_G^{-\frac{1}{2}}(i)Q_G^T(i)$ a change of coordinates that redresses the ellipsoid defined by the NL structure tensor into a sphere.

Finally, we define $\tilde{T}_G(i) = \max(\delta, |S_G\nabla_G u|(i))T_G(i)$ in the same manner than Section II, and our adaptive NL regularizer as follows:

$$\mathcal{R}_{J_G}(u) = \sum_{i\in\Omega} g_\delta(S_G\nabla_G u), \ S_G = \max(\sqrt{\delta}, \sqrt[4]{\lambda_1+\lambda_2}) \ \Lambda_G^{-\frac{1}{2}} Q_G^T.$$

D. Robust Estimation of the Non-Local Structure Tensor

Each iteration the NL structure tensor is estimated in two steps. We first obtain a noisy point-wise estimate $\hat{J}_G$ from the image u_{k-1}, and we then regularize it to obtain J_{G_k}.

As in the local setting, $\hat{J}$ is defined by the outer product of image gradients, as follows:

$$\hat{J}_G(i) = \nabla_G u(i,\cdot) \otimes \nabla_G u(i,\cdot) \quad \text{where} \quad \hat{J}_G(i) \in \mathbb{S}_+^{\nu_i}. \tag{37}$$

$\hat{J}_G$ is computed directly in terms of NL derivatives and is very sensitive to noise. To overcome this issue, we regularize it with a technique adapted to the NL structure tensor. As in any regularization model, we have a data and a smoothness term, but the smoothness term requires careful consideration.

The difficulty comes from the lack of global correspondence between components of $\hat{J}_G$ and spatial directions in Ω that is inherited from the NL gradient. Indeed, the direction of $\nabla_G u(i,j)$ depends on the relative position of nodes i and j in the image domain and is not coherent for all the pixels. Consequently, we cannot compare the value of the NL structure tensor in terms of its components, and regularization based on component-wise differential operators is meaningless. There is no reason, however, not to combine a data term on $\hat{J}_G$ with a smoothness term on a scalar function of J_G.

We consider log-Euclidean metrics in the positive semi-definite cone for both the data and the smoothness terms. This results in the following minimization in variable $L_G = \log J_G$

$$\min_{\substack{L_G \\ v=\mathrm{tr}[\hat{L}_G L_G]}} \sum_{i\in\Omega} g_\nu(\nabla_G v) + \frac{\beta}{2}\|L_G(i) - \hat{L}_G(i)\|_F^2, \tag{38}$$

where $\mathrm{tr}[\cdot]$ is the trace operator, $\hat{L}_G = \log(\hat{J}_G)$, and v is an affine approximation of the Frobenius norm of L_G, that is,

$$\|L_G(i)\|_F^2 = \mathrm{tr}[L_G(i) L_G(i)] \approx \mathrm{tr}[\hat{L}_G(i) L_G(i)]. \tag{39}$$

This approximation is introduced to make the minimization convex and the numerical algorithms fast. After minimization, the structure tensor is recovered by matrix exponentiation.

E. Enhanced Non-Local Image Reconstruction

Given the proposed regularizer $\mathcal{R}_{J_{G_k}}$ we reconstruct the image by solving the following minimization problem:

$$u_k = \arg\min_u \ \mathcal{R}_{J_{G_k}}(u) + \frac{\alpha}{2}\|Au - f\|_2^2. \tag{40}$$

V. Numerical Minimization of NL Method

As an extension of our local method, we can develop efficient minimization algorithms for the NL case by adapting the algorithms of Section III. To avoid repetition, we simply write here the form of the constrained minimizations, the resulting saddle-point problems, and each of the sub-minimizations.

The NL gradient is a linear operator and, by arranging the discretized image as a column vector, $\nabla_G u$ is computed by the matrix multiplication $D_G u$. Matrix D_G is derived from the weights of the graph's edges and is therefore sparse.

A. Reconstruction with NL Adaptive Regularizer

To reconstruct the image we need to solve

$$\min_u \sum_{i\in\Omega} g_\delta(S\nabla u) + \frac{\alpha}{2}\|Au - f\|_2^2, \tag{41}$$

which is re-formulated as the following convex minimization:

$$\min_{u,\ \boldsymbol{d}_G} \ F_G(\boldsymbol{d}_G) + \frac{\alpha}{2}\|Au - f\|_2^2 \ \text{ s.t } \ \boldsymbol{d}_G = K_G u, \tag{42}$$

where $F_G(\boldsymbol{d}) = \sum_{i\in\Omega}\ g_\delta(\boldsymbol{d})$, and K_G is the linear operator that combines graph-derivative and the change of variables S_G, that is, $K_G u(i) = S_G(i)\nabla_G u(i)$. With this notation, the associated saddle-point problem looks similar to the local case:

$$\min_u \max_{\boldsymbol{y}_G} \ \frac{\alpha}{2}\|Au - f\|_2^2 - F_G^*(\boldsymbol{y}_G) + \langle \boldsymbol{y}_G, K_G u\rangle. \tag{43}$$

The only major differences between the local and NL versions, (43) and (17), are the substitution of F by F_G and the linear operator K_G by K. Consequently, in the primal-dual Algorithm 1, we only need to update the $\boldsymbol{y}_G$-step that solves a proximal problem in F_G^*.

We use again Moreau's identity to rewrite the minimization problem in terms of F_G, this results in

$$\boldsymbol{y}_G^\star = \hat{\boldsymbol{y}}_G - \sigma \boldsymbol{d}_G^\star \quad \boldsymbol{d}_G^\star = \arg\min_{\boldsymbol{d}_G} \ F_G(\boldsymbol{d}_G) + \frac{\sigma}{2}\|\boldsymbol{d}_G - \frac{\hat{\boldsymbol{y}}_G}{\sigma}\|_{G,2}^2.$$

This problem is again decoupled for each pixel as follows:

$$\min_{\boldsymbol{d}_G(i)} \sum_i \left[g_\delta(\boldsymbol{d}_G(i)) + \frac{\sigma}{2}|\boldsymbol{d}_G(i) - \frac{\hat{\boldsymbol{y}}_G(i)}{\sigma}|_G^2 \right]. \tag{44}$$

It has the same form of the discretization of its local version (23), except that now the size of the vector $\boldsymbol{d}_G(i)$ depends on the node. Nevertheless, as the problem is solved independently for each node, we obtain the same closed-form solution (25). The dual variable $\boldsymbol{y}_G$ is then updated:

$$\boldsymbol{y}_G(i) = \frac{\hat{\boldsymbol{y}}_G(i)}{\max(|\hat{\boldsymbol{y}}_G(i)|_G, 1 + \sigma\delta)}. \tag{45}$$

The final procedure is summarized in Algorithm 3.

Initialize $u^0 = \mathbf{0},\ \boldsymbol{y}^0 = \mathbf{0},\ z^0 = u.$

Choose $\tau, \sigma > 0,\ \theta \in [0, 1].$

while $\|u^{t+1} - u^t\| > 1^{-4}$ **do**

$$\begin{aligned}
&\boldsymbol{y}_G^{t+1}(i) = \ \tfrac{\hat{\boldsymbol{y}}_G(i)}{max(1+\sigma\delta, |\hat{\boldsymbol{y}}_G(i)|_G)}, \quad \hat{\boldsymbol{y}}_G = \boldsymbol{y}_G^t + \sigma K_G z^t \\
&u^{t+1} = \tfrac{\hat{u}+\alpha f}{1+\tau\alpha}, \qquad\qquad\qquad \hat{u} = u^t - \tau K^* \boldsymbol{y}^{t+1} \\
&z^{t+1} = \ u^{t+1} + \theta(u^{t+1} - u^t)
\end{aligned}$$

Algorithm 3: *Efficient minimization of* (41) *for image denoising; deblurring and compressed sensing reconstructions update* u *with* (21).

B. Regularization of the NL Structure Tensor

To regularize the NL structure tensor, we solve

$$\min_{\substack{L_G \\ v = \mathrm{tr}[\hat{L}_G L_G]}} \ \sum_{i \in \Omega} g_\nu(\nabla_G v) + \frac{\beta}{2} \|L_G(i) - \hat{L}_G(i)\|_F^2 \tag{46}$$

by re-formulating it as the constrained minimization

$$\min_{L_G} \ F_G(\boldsymbol{p}_G) + \frac{\beta}{2} \|L_G - \hat{L}_G\|_{F,2}^2 \text{ s.t. } \boldsymbol{p}_G = P_G L_G, \tag{47}$$

where P_G is the linear operator that combines ∇_G with the node constraint on v, i.e., $P_G\, L_G(i) \ = \ \nabla_G \ \mathrm{tr}[\, \hat{L}_G(i)\ L_G(i)]$.

We observe that problem (47) has the same form as (42) if we susbtitute the linear operator K_G by P_G and $A = Id$. Consequently, we can directly adapt Algorithm 3 to this problem; the result is Algorithm 4.

Initialize $L_G^0 = z^0 = \log \hat{J}_G,\ \boldsymbol{y}^0 = \mathbf{0},\ \tau, \sigma > 0,\ \theta \in [0, 1]$

while $\|L_G^{t+1} - L_G^t\| > 1^{-4}$ **do**

$$\begin{aligned}
&\boldsymbol{y}_G^{t+1}(i) = \ \tfrac{\hat{\boldsymbol{y}}_G(i)}{max(1+\sigma\nu, |\hat{\boldsymbol{y}}_G(i)|_G)}, \quad \hat{\boldsymbol{y}}_G = \boldsymbol{y}_G^t + \sigma P_G z^t \\
&L_G^{t+1} = \ \tfrac{L_G^t - \tau P_G^* \boldsymbol{y}_G^{t+1} + \beta \hat{L}_G}{1+\tau\beta} \\
&z^{t+1} = \ L_G^{t+1} + \theta(L_G^{t+1} - L_G^t)
\end{aligned}$$

Algorithm 4: *Efficient minimization of* (38).

VI. Numerical Experiments

We present both quantitative and qualitative results on image denoising, deblurring, and compressed-sensing reconstruction. In denoising we add Gaussian noise to the images to define our measurements

f; in deblurring f is the convolution of the original image with a Gaussian kernel of size σ; while in compressed sensing f are m Fourier measurements radially sampled in the frequency domain.

We compare our local method to classic TV regularization, as our technique can be considered as a generalization of it. To compare our model to anisotropic diffusion based on the structure tensor, we use the anisotropic regularizer recently proposed by [13]. The parameters of each model (α for TV, α and the size of the Gaussian kernel in [13]), and $\alpha, \beta, \gamma, \mu$ for our model) have been optimized manually for the best peak SNR, which is the metric used in our evaluation.

For the non-local method, we compare its performance to the state-of-the-art BM3D [43] and non-local TV of [44], where the weights of the NL regularizers are computed from the reconstructions obtained with the respective local versions. Due to its formulation, BM3D can only be applied to image denoising – and to some extend to deblurring – and we restrict our comparison to this application; this is in fact the main drawback of BM3D compared to regularization, which can be applied to any inverse problem.

A. Results

We present our results in Figures 1-3 and Tables I-III, and we analyze them by answering several questions.

1) Isotropic vs Anisotropic Regularization: The main difference between our model and classic TV and Dirichlet regularizers lies in its anisotropy. Anisotropy is also a feature of natural images, both of highly oriented images like fingerprints and of textured images like *Barbara*; this results in an improved performance of our regularizer in comparison to classic TV: 0.55 dB in average for denoising, 0.8 dB for deblurring, and 1.4 dB for CS reconstruction. Our experiments show that all the images benefit from the use of an adaptive regularizer, with the improvement is limited to textured images.

2) Effects of non-differentiability in $\mathcal{R}_J$: Compared to regularizers based on ℓ_1 penalties, like TV or [13], our model allows for classic diffusion when the image variation is small by the use of the robust Huber norm. This eliminates the staircase effect associated with ℓ_1-minimization models that characterizes total variation. Such an effect is visible in *Lena*'s shoulder or the sky of *boat* and *castle* images.

3) Robust Estimation of Structure Tensor: Another point that differentiates our model from regularization schemes based on the structure tensor is the estimation of this tensor. While [1], [10], [11], [13] use Gaussian smoothing of (5) to locally average the components of this tensor, our averaging takes into account the local structure of the image, by formulating the estimation as a tensor denoising problem, and the nature of the tensor measurements by the use of log-Euclidean metrics. While the benefits of log-Euclidean metrics depend on the application (it improves reconstruction for image deblurring, but degrades

for denosing and compressed sensing), our estimation of the structure tensor as an SDP optimization is beneficial: an average gain of 0.85 dB over [13] in compressed sensing, 0.5 in deblurring, and 0.06 in denoising. Our model is in fact more robust for more ill-posed problems, always outperforming TV and [13] in compressed sensing and deblurring, while being comparable to it [13] in denoising for high levels of noise.

4) Local vs Non-Local Method: Figures 2-3 compare our local and non-local regularizers and shows how non-local regularization visually improves image reconstruction in compressed sensing, not only for textured images. This improvement does not extend to the other applications.

5) Use of geometry in non-local methods: Compared to non-local regularizes, our model improves signal reconstruction by exploiting the geometric information of the non-local graph. While non-local TV defines an isotropic regularizer in the graph of image patches, our non-local model is associated with an anisotropic diffusion operator that exploits the geometry of the image in the non-local graph. As in the local case, our anisotropic model outperforms its isotropic counterpart, the NL TV of [44], in all the denoising, deblurring, and compressed-sensing experiments.

For image denoising we also compare our model to BM3D [43], where multiple image patches collaboratively denoise each image pixel. Although BM3D outperforms the rest of techniques for image denoising, its formulation does not generalize to linear inverse problems. Our regularizer substitutes NL TV in any variational model and is therefore more general.

B. Discussion and Limitations

Our model relies on the structure tensor to estimate the image geometry and improve regularization, as a result, it only outperforms traditional TV if the structure tensor can be reliably estimated from u^0. For this reason, as the data is degraded – by increasing additive noise in denoising, increasing the size of the blurring kernel in deblurring, or reducing the measurements in compressed sensing– the relative performance of our approach degrades. An example of this behavior appears in Table I for denoising, where our model and [13] become comparable as the noise level increases.

The limitation of the NL method stems from the size of variables derived from differential operator and the accuracy of the weights that define the NL graph. Indeed, as NL methods define a gradient vector with as many components as NL neighbors, $\nabla u(i)$ has ν_i components for each pixel and its associated structure tensor ν_i^2; with ν_i in the range $5-9$ and 10^4-10^5 pixels per image, the proposed NL method has large memory requirements. The second limitation stems from the dependency of the NL weights (35) on the unknown image, that we overcome by computing the weights from an approximate image

estimate. In our experiments, this strategy is very successful for CS, where our NL technique outperforms the local one, but partially fails in image deblurring and denoising, where our NL technique outperforms NL TV but fails to improve our local reconstruction. This is a general limitation of the NL regularization framework of [3], [44].

In terms of computation, our local method is in average 3-5 times slower than TV regularization and 3-4 times slower than [13], with performance depending on the application and experimental setting. The non-local version is 6-8 times slower than NL-TV, with 70-80% of the time dedicated to the estimation of the high-dimensional NL structure tensor.

σ noise	5%				7.5%				10%			
	TV	[13]	ours	logE	TV	[13]	ours	logE	TV	[13]	ours	logE
lena	16.84	17.46	**17.99**	15.13	14.48	15.11	**15.58**	12.06	13.02	13.62	**13.89**	10.61
barbara	17.09	17.77	**18.01**	15.26	14.77	15.44	**15.61**	12.24	13.13	13.80	**13.84**	10.50
castle	17.90	18.34	**18.43**	16.88	15.65	**16.10**	16.09	13.82	14.15	14.56	**14.46**	12.02
boat	15.14	15.52	**15.60**	13.62	12.84	13.23	**13.26**	11.09	11.52	11.82	**11.83**	9.16
fingerprint	12.24	12.46	**12.58**	11.99	9.23	9.54	**9.71**	9.14	7.31	7.67	**7.85**	7.25
baboon	13.52	**13.77**	**13.77**	12.72	11.23	**11.55**	11.43	10.02	9.76	**10.04**	9.91	8.11
monarch	13.71	13.98	**14.33**	13.10	11.01	11.35	**11.65**	10.12	9.19	9.52	**9.72**	7.88

TABLE I: *Denoising experiments with different levels of Gaussian noise σ. The first two columns show the peak SNR of TV and the anisotropic regularizer of [13], the last two our model with Euclidean (ours) or log-Euclidean (logE) metrics.*

VII. Conclusions

We propose an adaptive regularizer for image reconstruction inspired by classic diffusion PDEs. The new regularizer adapts to the geometry of each image and naturally extends to non-local operators for texture recovery. Compared to diffusion PDEs, our variational formulation is able to reconstruct non-smooth images and allows for modern optimization techniques. In particular, we develop efficient algorithms for image denoising, deblurring, and compressed-sensing reconstruction by solving a sequence of proximal convex problems with analytical closed-form solutions. Our experiments show a consistent gain in accuracy over classic regularization in both local and non-local settings.

blurr	$\sigma = 1$						$\sigma = 1.5$						$\sigma = 2$					
	local methods				NL methods		local methods				NL methods		local methods				NL methods	
	TV	[13]	our	logE	[44]	NL our	TV	[13]	our	logE	[44]	NL our	TV	[13]	our	logE	[44]	NL our
lena	6.50	6.90	7.76	**8.47**	6.56	**6.82**	6.42	6.88	7.52	**7.84**	6.54	**6.86**	6.33	6.79	6.79	**7.25**	6.59	**6.83**
barbara	6.88	7.30	8.58	**9.34**	7.10	**7.26**	6.90	7.33	8.30	**8.64**	7.12	**7.29**	6.84	7.28	7.73	**7.90**	7.12	**7.13**
castle	9.37	9.62	10.52	**11.08**	9.19	**9.25**	9.40	9.63	10.23	**10.41**	9.32	**9.43**	9.34	9.59	9.77	**9.81**	**9.36**	9.36
boat	6.38	6.69	7.62	**8.40**	6.52	**6.78**	6.34	6.69	7.28	**7.66**	6.48	**6.75**	6.28	6.64	6.93	**7.05**	6.40	**6.62**
fingerprint	2.55	2.58	2.85	**3.10**	2.52	**2.54**	2.55	2.59	2.71	**2.74**	2.55	**2.58**	2.54	2.58	**2.59**	2.58	2.56	**2.56**
baboon	6.08	6.25	7.21	**7.72**	6.03	**6.26**	6.05	6.21	6.85	**7.04**	6.02	**6.25**	6.00	6.14	6.39	**6.44**	5.99	**6.11**
monarch	3.75	3.91	4.80	**5.56**	3.92	**4.12**	3.68	3.84	4.18	**4.45**	3.82	**3.97**	3.57	**3.75**	3.59	3.67	**3.67**	3.60

TABLE II: *Peak SNR in deblurring with Gaussian kernels of width σ. The first four columns compare our local method – with Euclidean and log-Euclidean (logE) metrics – to TV and the anisotropic regularizer of [13], the last two columns compare our NL model to NL TV [2]. Our local method outperforms TV and [13], while our NL technique outperforms TV and NL TV.*

ACKNOWLEDGEMENT

The authors would like to thank the authors of [13] for sharing their code. Research supported in part by ONR N000141110863, AFOSR FA9550-12-1-0364. V. Estellers is supported by the Swiss National Science Foundation under grant P2ELP2_148890.

REFERENCES

[1] J. Weickert, *Anisotropic diffusion in image processing*. Stuttgart: Teubner Stuttgart, 1998, vol. 1.

[2] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Model. Simul.*, vol. 7, no. 3, pp. 1005–1028, 2008.

[3] D. Zhou and B. Schölkopf, "Regularization on Discrete Spaces," in *Pattern Recognition. Lect. Notes Comput. Sci.*, ser. Lecture Notes in Computer Science, W. Kropatsch, R. Sablatnig, and A. Hanbury, Eds. Springer, 2005, vol. 80, pp. 361–368.

[4] O. Scherzer, *Handbook of Mathematical Methods in Imaging: Vol. 1*, ser. Springer Reference. Springer, 2011.

[5] N. Paragios, Y. Chen, and O. D. Faugeras, *Handbook of Mathematical Models in Computer Vision*. Springer, 2006.

[6] O. Scherzer and J. Weickert, "Relations Between Regularization and Diffusion Filtering," *J. Math. Imaging Vis.*, vol. 63, pp. 43–63, 2000.

[7] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D Nonlinear Phenom.*, vol. 60, no. 1-4, pp. 259–268, Nov. 1992.

[8] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.

m/n	12%						18%						24%					
	local methods				NL methods		local methods				NL methods		local methods				NL methods	
	TV	[13]	our	logE	[44]	NL our	TV	[13]	our	logE	[44]	NL our	TV	[13]	our	logE	[44]	NL our
lena	10.32	10.80	11.47	**11.54**	10.73	**11.84**	15.97	17.15	**19.00**	17.09	17.40	**19.11**	20.13	21.78	**23.25**	20.26	22.15	**23.07**
barbara	10.12	10.62	**11.78**	11.55	10.50	**11.92**	15.42	16.58	**18.02**	17.04	16.85	**18.51**	19.57	21.44	**22.64**	20.47	21.52	**22.80**
castle	12.19	12.47	**12.66**	12.05	12.27	**12.58**	16.61	17.08	**18.30**	16.75	17.30	**18.28**	19.82	20.51	**21.57**	19.65	20.97	**21.79**
boat	8.69	9.06	**9.35**	9.17	9.49	**9.50**	13.12	13.78	**14.56**	13.51	13.91	**14.80**	16.32	17.10	**18.24**	16.37	17.16	**18.04**
fingerprint	2.75	3.05	**3.06**	3.05	2.94	**3.16**	3.83	4.30	4.38	**4.44**	4.18	**5.20**	5.17	5.81	5.86	**6.78**	5.93	**7.99**
baboon	6.78	7.00	7.11	**7.17**	6.86	**7.26**	9.91	10.40	10.37	**10.47**	10.12	**10.64**	11.64	**12.21**	12.12	12.11	11.92	**12.33**
monarch	4.66	5.01	**5.37**	5.34	4.94	**5.67**	9.92	10.33	**12.57**	10.54	10.68	**12.66**	14.39	15.11	**17.49**	14.63	15.55	**17.03**

TABLE III: *Peak SNR in compressed-sensing reconstruction with sampling ratio m/n. The first four columns compare our local method – with Euclidean and log-Euclidean (logE) metrics – to TV and the anisotropic regularizer of [13], the last two columns compare our NL model to NL TV [2]. Our local method outperforms TV and [13], while our NL technique obtains the best overall performance from both local and NL methods.*

σ noise	5%			7.5%			10%		
	[44]	ours	BM3D	[44]	ours	BM3D	[44]	ours	BM3D
lena	17.51	**17.71**	18.62	15.27	**15.50**	16.15	13.74	**14.04**	14.74
barbara	17.82	**18.02**	18.87	15.47	**15.69**	16.51	13.74	**13.91**	14.94
castle	17.96	**18.12**	19.36	15.72	**15.75**	16.94	**14.17**	13.90	15.35
fingerprint	13.06	**13.12**	13.40	10.39	**10.53**	11.00	8.58	**8.77**	9.42
baboon	13.52	**13.66**	14.16	11.24	**11.41**	11.93	9.72	**9.88**	10.40
monarch	13.82	**14.06**	14.92	11.21	**11.50**	12.26	9.39	**9.70**	10.42

TABLE IV: *Peak SNR in denoising with Gaussian noise. The first two columns compare our NL method to NL-TV [44], the last column show the results from BM3D. Our NL method outperforms the general NL variational regularizer [44], but does worse than the denoising-specific technique BM3D [43].*

[9] J. Weickert, B. M. t. H. Romeny, and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering." *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 398–410, Jan. 1998.

[10] M. Grasmair and F. Lenzen, "Anisotropic Total Variation Filtering," *Appl. Math. Optim.*, vol. 62, no. 3, pp. 323–339, May 2010.

[11] A. Roussos and P. Maragos, "Tensor-based Image Diffusions Derived from Generalizations of the Total Variation and Beltrami Functionals," *IEEE Int. Conf. Image Process.*, pp. 4141–4144, 2010.

[12] N. Sochen, R. Kimmel, and R. Malladi, "A general framework for low level vision." *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 310–318, Jan. 1998.

[13] S. Lefkimmiatis, A. Roussos, M. Unser, and P. Maragos, "Convex Generalizations of Total Variation based on the Structure Tensor with Applications to Inverse Problems," *Scale Sp. Var. Methods Comput. Vis.*, vol. 7893, pp. 48–60, 2013.

[14] Y. Dong and M. Hintermüller, "Multi-scale Total Variation with Automated Regularization Parameter Selection for Color Image Restoration," in *Scale Sp. Var. Methods Comput. Vis. SE - 23*, ser. Lecture Notes in Computer Science, X.-C. Tai, K. Mø rken, M. Lysaker, and K.-A. Lie, Eds. Springer Berlin Heidelberg, 2009, vol. 5567, pp. 271–281.

[15] G. Gilboa, N. Sochen, and Y. Y. Zeevi, "Variational denoising of partly textured images by spatially varying constraints," pp. 2281–2289, 2006.

[16] M. Grasmair, "Locally Adaptive Total Variation Regularization," in *Scale Sp. Var. Methods Comput. Vis. SE - 28*, ser. Lecture Notes in Computer Science, X.-C. Tai, K. Mø rken, M. Lysaker, and K.-A. Lie, Eds. Springer Berlin Heidelberg, 2009, vol. 5567, pp. 331–342.

[17] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM J. Imaging Sci.*, vol. 3, no. 3, pp. 492–526, 2010.

[18] I. Bayram and M. E. Kamasak, "Directional Total Variation," *IEEE Signal Process. Lett.*, vol. 19, no. 12, pp. 781–784, Dec. 2012.

[19] P. J. Huber, "Robust Estimation of a Location Parameter," pp. 73–101, 1964.

[20] P. Blomgren and T. F. Chan, "Color TV: total variation methods for restoration of vector-valued images." *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 304–309, Jan. 1998.

[21] M. Moakher, "A Differential Geometric Approach to the Geometric Mean of Symmetric Positive-Definite Matrices," *SIAM J. Matrix Anal. Appl.*, vol. 26, no. 3, pp. 735–747, Jan. 2005.

[22] P. G. Batchelor, M. Moakher, D. Atkinson, F. Calamante, and A. Connelly, "A rigorous framework for diffusion tensor calculus." *Magn. Reson. Med.*, vol. 53, no. 1, pp. 221–225, Jan. 2005.

[23] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-Euclidean metrics for fast and simple calculus on diffusion tensors." *Magn. Reson. Med.*, vol. 56, no. 2, pp. 411–421, Aug. 2006.

[24] G. Rosman, Y. Wang, X. X.-C. Tai, R. Kimmel, and A. Bruckstein, "Fast Regularization of Matrix-Valued Images," in *Effic. Algorithms Glob. Optim. Methods Comput. Vis. SE - 2*, ser. Lecture Notes in Computer Science, A. Bruhn, T. Pock, and X.-C. Tai, Eds. Springer Berlin Heidelberg, 2014, pp. 19–43.

[25] R. Courant, K. Friedrichs, and H. Lewy, "On the Partial Difference Equations of Mathematical Physics," *IBM J. Res.*, vol. 3, no. March, pp. 215–234, 1967.

[26] A. Chambolle, "An Algorithm for Total Variation Minimization and Applications," *J. Math. Imaging Vis.*, vol. 20, no. 1/2, pp. 89–97, Jan. 2004.

[27] T. F. Chan, G. H. Golub, and P. Mulet, "A Nonlinear Primal-Dual Method for Total Variation-Based Image Restoration," *SIAM J. Sci. Comput.*, vol. 20, no. 6, p. 1964, 1999.

[28] M. Zhu, S. J. Wright, and T. F. Chan, "Duality-based algorithms for total-variation-regularized image restoration," *Comput. Optim. Appl.*, vol. 47, no. 3, pp. 377–400, Dec. 2010.

[29] A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems withApplications to Imaging," *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, Dec. 2010.

[30] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imaging*, pp. 1–24, 2008.

[31] T. Goldstein and S. Osher, "The Split Bregman Method for l1 Regularized Problems," *SIAM J. Imaging Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

[32] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman Iterative Algorithms for l1-Minimization with Applications to Compressed Sensing," *SIAM J. Imaging Sci.*, vol. 1, no. 1, pp. 143–168, 2008.

[33] C. Wu and X.-C. Tai, "Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models," *SIAM J. Imaging Sci.*, vol. 3, no. 3, pp. 300–339, 2010.

[34] A. Szlam, M. Maggioni, and R. R. Coifman, "A general framework for adaptive regularization based on diffusion processes on graphs," *Yale Tech. Rep.*, 2006.

[35] G. Peyré, S. Bougleux, and L. Cohen, "Non-local regularization of inverse problems," *Eur. Conf. Comput. Vis.*, pp. 57–68, 2008.

[36] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *SIAM J. Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.

[37] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal Discrete Regularization on Weighted Graphs: A Framework for Image and Manifold Processing," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1047–1060, 2008.

[38] G. Gilboa and S. Osher, "Nonlocal Linear Image Regularization and Supervised Segmentation," *Multiscale Model. Simul.*, vol. 6, no. 2, pp. 595–630, 2007.

[39] X. Bresson and T. F. Chan, "Non-local unsupervised variational image segmentation models," *UCLA CAM Rep.*, vol. 08-67, pp. 1–34, 2008.

[40] X. Zhang, M. Burger, X. Bresson, and S. Osher, "Bregmanized nonlocal regularization for deconvolution and sparse reconstruction," *SIAM J. Imaging Sci.*, vol. 3, no. 3, pp. 253–276, 2010.

[41] V. Estellers, J.-P. Thiran, and X. Bresson, "Enhanced Compressed Sensing Recovery with Level Set Normals," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2611–2627, Mar. 2013.

[42] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 53–94, 2006.

[43] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering." *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Jan. 2007.

[44] G. Gilboa and S. Osher, "Nonlocal Operators with Applications to Image Processing," *Multiscale Model. Simul.*, vol. 7, no. 3, p. 1005, 2009.

(a) noisy image $\sigma_n = 0.05$

(b) TV reconstrucion

(c) anisotropic regularizer [13]

(d) our local method

(e) noisy image $\sigma_n = 0.05$

(f) TV reconstruction

(g) anisotropic regularizer [13]

(h) our local method

(i) noisy image $\sigma_n = 0.05$

(j) TV reconstruction

(k) anisotropic regularizer [13]

(l) our local method

(m) noisy image $\sigma_n = 0.075$

(n) TV reconstruction

(o) anisotropic regularizer [13]

(p) our local method

(q) noisy image $\sigma_n = 0.075$

(r) TV reconstruction

(s) anisotropic regularizer [13]

(t) our local method

Fig. 1: Image denosing with noise level $\sigma_n = 0.05 - 0.075$.

(a) Fourier back projection of f

(b) TV reconstruction

(c) anisotropic regularizer [13]

(d) our local method

(e) NL TV reconstruction

(f) our NL method

(g) Fourier back projection of f

(h) TV reconstruction

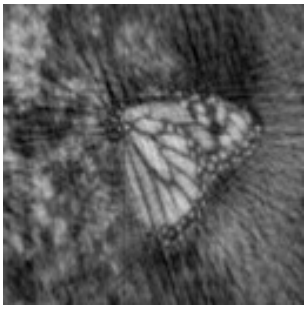
(i) anisotropic regularizer [13]

(j) our local method

(k) NL TV reconstruction

(l) our NL method

Fig. 2: Compressed-sensing reconstruction from 18% of Fourier samples.



(a) Fourier back projection of f

(b) TV reconstruction

(c) anisotropic regularizer [13]

(d) our local method

(e) NL TV reconstruction

(f) our NL method

Fig. 3: Compressed-sensing reconstruction of *monarch* from 18% of Fourier samples.