A Decomposition Framework for Image Denoising Algorithms

Gabriela Ghimpețeanu, Thomas Batard, Marcelo Bertalmío and Stacey Levine

***Abstract*—In this paper we consider an image decomposition model that provides a novel framework for image denoising. The model computes the components of the image to be processed in a moving frame that encodes its local geometry (directions of gradients and level-lines). Then, the strategy we develop is to denoise the components of the image in the moving frame in order to preserve its local geometry, which would have been more affected if processing the image directly. Experiments on a whole image database tested with several denoising methods show that this framework can provide better results than denoising the image directly, both in terms of PSNR and SSIM [27] metrics.**

***Index Terms*—Image denoising, local variational method, patch-based method, differential geometry.**

I. Introduction

DENOISING an image is a fundamental task for correcting defects produced during the acquisition process of a real-world scene and its reproduction on a display, due to physical and technological limitations (see e.g. [5] for more details). It can also be useful as a pre-processing stage in order to improve the results of higher level applications.

The problem of removing the noise of an image while preserving its main features (edges, textures, colors, contrast, etc.) has been extensively investigated over the last two decades and several types of approaches have been developed. The total variation-based denoising method of Rudin el al. [25] had a great impact in the imaging community and has inspired a large amount of variational formulations for image denoising. Years after the model of Rudin et al., a novel approach for image denoising based on the comparison of pixel neighborhoods (patches) was proposed simultaneously by Awate and Whitaker [1] with the UINTA algorithm and Buades et al. [9] with the Non-Local Means (NLM) algorithm. To a great extent, these patch based methods outperformed the denoising models that existed at that time. Since then, a number of patch-based methods have been developed, comprising the majority of the current state-of-the-art denoising methods. We refer the reader to the paper of Lebrun et al. [17] for a complete description of the denoising problem, as well as a detailed analysis and comparison of state-of-the-art denoising methods. It has been shown by Levin and Nadler [20] and Chaterjee and Milanfar [12] that the current state-of-the-art denoising methods are close to optimal when applied to natural images. Nonetheless, there is still room for improvement in several directions. For instance, while these methods manage to correctly remove most of the noise, they tend to not properly recover some of the image details. These methods also primarily deal with additive Gaussian noise, whereas for many images the noise model is unknown; in such cases, there is still ample room for improvement (see Lebrun et al. [19] and references therein for blind denoising algorithms). Our proposal in this paper is to develop a strategy to improve any image denoising technique by more carefully taking into account the local geometry (direction of gradients and level-lines) of the image to process.

Previous works of the authors: Motivated by the construction of a Fourier theory for n-channel images that would involve the local geometry of the image, Batard and Berthier [2] proposed the following approach: construct an orthonormal moving frame of $\mathbb{R}^{n+2}$ over the image domain where the first two vector fields are tangent to the image graph and the n remaining vector fields are normal to that surface. Then, representing the image in this n+2 dimensional moving frame, compute the standard 2D Fourier transform of each of the components, apply Gaussian kernels of different sizes in the Fourier domain, and then project back. This essentially applies a Euclidean heat diffusion on each of the components, obtaining a heat diffusion on the original image that preserves its local geometry throughout the diffusion process. Later on, Batard and Bertalmío [3],[4] followed this strategy for the purpose of image denoising. More precisely, instead of applying a heat equation on the components of the image in the moving frame, they applied a denoising method and compared the results with applying that method directly on the image. In [3], they dealt with the vectorial extension of the total variation-based denoising method of Rudin et al. [25] proposed by Blomgren and Chan [7], and in [4] they dealt with the so-called vectorial total variation (VTV) of Bresson and Chan [8]. In both cases, they showed that this methodology can improve the standard approach in terms of PSNR and Q-Index [26] metrics, the latter being more related to perception than the PSNR.

A comparable framework was adopted by Bertalmío and Levine [6]. Based on the observation that the curvature of the level-lines of a gray-level image is less affected by the noise than the intensity values of the image (assuming that the image has been corrupted by additive Gaussian noise), they conjectured that it can be easier to recover the curvature of the clean image than the clean image itself. Experiments

G. Ghimpețeanu, T. Batard and M. Bertalmío are with the Department of Information and Communication Technologies, University Pompeu Fabra, 08018 Barcelona, Spain (e-mail: {gabriela.ghimpeteanu;thomas.batard;marcelo.bertalmio}@upf.edu)

S. Levine is with the Department of Mathematics and Computer Science, Duquesne University, Pittsburgh, PA 15282 USA (e-mail: sel@mathcs.duq.edu).

involving 4 different denoising methods: TV denoising performed through gradient descent [25] and Bregman iterative algorithm [22], orientation matching using smooth unit tangents [15], NLM [9] and BM3D [13], confirmed that applying a denoising method on the curvature of the noisy image and then reconstructing an image from the denoised curvature provides a better approximation to the clean image (in terms of PSNR and Q-Index metrics) than applying this denoising method directly on the noisy image.
A similar approach for image denoising is the one of Lysaker et al. [21] who first smooth the unit normal field to the image level-lines, then construct a denoised image whose unit normal field would match the smooth field. Some similar approaches for denoising that have been inspired by the work of Lysaker et al. can be found in [15],[22],[23].

Contribution: This paper, a substantial extension of our earlier work in [14], shows that given a denoising method we can obtain better, cleaner results by denoising the components of an image in a moving frame (as in [2],[3],[4]), compared to what we would get by denoising the image directly. We formally prove that, along image contours, the PSNR of the components is higher than that of the image, which would explain the ability of our framework to better preserve image details regardless of the particular image denoising technique that is applied. We have been able to improve three denoising methods of different types: a local variational method (VTV, [8]), a patch-based method (NLM, [9]), and a method combining a patch-based approach with a filtering in spectral domain approach (BM3D, [13]); the improvement is both in terms of PSNR and SSIM [27] metrics, and for grayscale and color images over a standard image database, demonstrating the consistency of our strategy. Compared to the curvature-based approach in [6] or the above mentioned methods based on denoising fields ([21],[15],[22],[23]), the asset of our framework holds in the simplicity of the reconstruction step, consisting simply in applying a matrix transform to the denoised components to obtain the denoised image, instead of a second or third order PDE evolution equation as in the previous techniques.

Outline: The outline of the article is the following. In section 2 we remind the reader of our image decomposition model for gray-level and multi-channel images. In section 3 we prove that our approach is suitable for image denoising by showing, both theoretically and numerically, that the PSNRs of the components of the noisy image in a well-chosen frame are higher than the PSNR of the noisy image itself on image contours. Finally, in section 4, we test our framework with three denoising methods: VTV minimization, NLM and BM3D. The results confirm the suitability of our approach since they show that it improves each aforementioned denoising method with respect to the PSNR and SSIM metrics on a whole database of gray-level and color images tested with different noise levels.

II. Image decomposition in a moving frame

A. The gray-level case

Let $I\colon \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}$ be a gray-level image, and (x,y) be the standard coordinate system of $\mathbb{R}^2$. We denote by I_x resp. I_y the derivative of I with respect to x resp. y, and by ∇I the gradient of I.

Our image decomposition model for I is a two-stages approach: first, we construct an orthonormal moving frame $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N})$ of $(\mathbb{R}^3, \| \, \|_2)$ over Ω that encodes the local geometry of I. Then, we compute the components (J^1, J^2, J^3) of the $\mathbb{R}^3$-valued function $(0,0,I)$ in that moving frame.

More precisely, we consider a scaled version μI of I, for $\mu \in]0,1]$, and its graph, which is the surface S in $\mathbb{R}^3$ parametrized by

$$\psi\colon (x,y) \longmapsto (x, y, \mu\, I(x,y)) \tag{1}$$

The orthonormal moving frame $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N})$ we consider is the following: the vector field $\boldsymbol{Z_1}$ is tangent to the surface S and indicates the direction of the steepest slope at each point of S; the vector field $\boldsymbol{Z_2}$ is tangent to S and indicates the direction of the lowest slope at each point of S. It follows that $\boldsymbol{N}$ is normal to the surface since we require $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N})$ to be orthonormal.

The moving frame $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N})$ can be constructed as follows. Let $\boldsymbol{z_1} = (\mu I_x, \mu I_y)^T$ be the gradient of μI and $\boldsymbol{z_2} = (-\mu I_y, \mu I_x)^T$ indicating the direction of the level-lines of μI. On homogeneous regions of I, i.e. at pixel locations (x,y) where $I_x(x,y) = I_y(x,y) = 0$, we define $\boldsymbol{z_1} = (1,0)^T$ and $\boldsymbol{z_2} = (0,1)^T$. Then, $\boldsymbol{Z_1}$ and $\boldsymbol{Z_2}$ are given by the following expressions

$$\boldsymbol{Z_i} = \frac{d\psi(\boldsymbol{z_i})}{\|d\psi(\boldsymbol{z_i})\|_2}, \qquad i = 1,2 \tag{2}$$

where $d\psi$ stands for the differential of ψ, which maps vector fields on Ω to tangent vector fields of S. The expression of the unit normal $\boldsymbol{N}$ is then obtained as the vectorial product between $\boldsymbol{Z_1}$ and $\boldsymbol{Z_2}$.

The explicit expressions of the vector fields $\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N}$ are given by the matrix field

$$P = \begin{pmatrix} \dfrac{I_x}{\sqrt{|\nabla I|^2(1+\mu^2|\nabla I|^2)}} & \dfrac{-I_y}{|\nabla I|} & \dfrac{-\mu\, I_x}{\sqrt{1+\mu^2|\nabla I|^2}} \\[2ex] \dfrac{I_y}{\sqrt{|\nabla I|^2(1+\mu^2|\nabla I|^2)}} & \dfrac{I_x}{|\nabla I|} & \dfrac{-\mu\, I_y}{\sqrt{1+\mu^2|\nabla I|^2}} \\[2ex] \dfrac{\mu|\nabla I|^2}{\sqrt{|\nabla I|^2(1+\mu^2|\nabla I|^2)}} & 0 & \dfrac{1}{\sqrt{1+\mu^2|\nabla I|^2}} \end{pmatrix}, \tag{3}$$

where the coordinates of the vector field $\boldsymbol{Z_1}$ are given in the first column, the coordinates of $\boldsymbol{Z_2}$ in the second column, and the coordinates of $\boldsymbol{N}$ in the third column.

Fig. 1 illustrates the moving frames $(\boldsymbol{z_1}, \boldsymbol{z_2})$ and $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N})$ aforementioned for a simple image. The left image shows the moving frame $(\boldsymbol{z_1}, \boldsymbol{z_2})$ at two points p and q of the domain Ω, and the right image shows the induced moving frame $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N})$ attached to the surface S at the points $\psi(p)$ and $\psi(q)$.

Denoting by $(\boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3})$ the orthonormal frame of $(\mathbb{R}^3, \| \, \|_2)$, where $\boldsymbol{e_1} = (1,0,0)$, $\boldsymbol{e_2} = (0,1,0)$, $\boldsymbol{e_3} = (0,0,1)$, the matrix P in (3) is nothing but the frame change field from

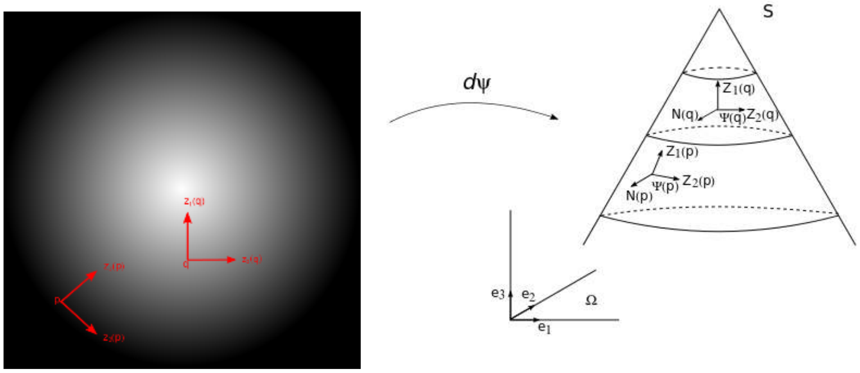

Fig. 1. Moving frame encoding the local geometry of a gray-level image. Left: original gray-level image and a moving frame $(\boldsymbol{z_1}, \boldsymbol{z_2})$ indicating the direction of the gradient and the level-line of the image at two points p and q of the image domain Ω. Right: the orthonormal moving frame $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N})$ of $(\mathbb{R}^3, \| \, \|_2)$ over Ω indicating the direction of the steepest and lowest slopes of the surface S, for some smoothing parameter μ, at the points $\psi(p)$ and $\psi(q)$.

$(\boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3})$ to $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N})$, meaning that the components of the $\mathbb{R}^3$-valued function $(0, 0, I)$ in the new frame, denoted by (J^1, J^2, J^3), are given by

$$\begin{pmatrix} J^1 \\ J^2 \\ J^3 \end{pmatrix} = P^{-1} \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix}. \tag{4}$$

Computing formula (4) gives that the component J^2 is identically 0.

Fig. 2 shows the gray-level image "Lena" and its components J^1 and J^3 for $\mu = 0.05$. We observe that the component J^1 contains the edges and textures of the image, which was expected according to formula (4) since the third component of the vector field $\boldsymbol{Z_1}$ is determined by the image gradient's norm, which is high on edges and textures. The component J^3 is similar to the original image from which the gradient's norm has been subtracted.

Finally, the parameter μ can be viewed as a smoothing parameter for the moving frame encoding the local geometry of the original image I. In fact, it turns out that μ plays a key role in our strategy for image denoising, as shown in the next sections.

B. The multi-channel case

We aim at extending the image decomposition model of Sect. II.A from gray-level to n-channel images $I = (I^1, \cdots, I^n) \colon \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}^n$, $n > 1$, by following a similar approach: first, we construct an orthonormal moving frame $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N_1}, \cdots, \boldsymbol{N_n})$ of $(\mathbb{R}^{n+2}, \| \, \|_2)$ over Ω that encodes the local geometry of I. Then, we compute the components $(J^1, J^2, \cdots, J^{n+2})$ of the $\mathbb{R}^{n+2}$-valued function $(0, 0, I^1, \cdots, I^n)$ in that moving frame.

As in the gray-level case, the first step consists in considering a scaled version μI of I, for $\mu \in]0, 1]$, and its graph, which is the surface S in $\mathbb{R}^{n+2}$ parametrized by

$$\psi \colon (x, y) \longrightarrow (x, y, \mu I^1(x, y), \cdots, \mu I^n(x, y)) \tag{5}$$

The moving frame $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N_1}, \cdots, \boldsymbol{N_n})$ we consider is then the following: the vector field $\boldsymbol{Z_1}$ is tangent to the surface S and indicates the direction of the steepest slope at each point of S; the vector field $\boldsymbol{Z_2}$ is tangent to S and indicates the direction of the lowest slope at each point of S, and $\boldsymbol{N_1}, \cdots, \boldsymbol{N_n}$ are normals to the surface. Note that, unlike the gray-level case, there is an infinite number of unit normals to the surface.

The moving frame $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N_1}, \cdots, \boldsymbol{N_n})$ can be constructed as follows. As in the gray-level case, $\boldsymbol{Z_1}$ and $\boldsymbol{Z_2}$ can be recovered from the directions $\boldsymbol{z_1}$ and $\boldsymbol{z_2}$ of highest and lowest variations of the scaled image μI under the map (2), these latter being the eigenvectors of the structure tensor associated to μI, which is given by

$$\begin{pmatrix} \sum_{k=1}^{n} (\mu I_x^k)^2 & \sum_{k=1}^{n} \mu^2 I_x^k \, I_y^k \\ \sum_{k=1}^{n} \mu^2 I_x^k \, I_y^k & \sum_{k=1}^{n} (\mu I_y^k)^2 \end{pmatrix} \tag{6}$$

On homogeneous regions, i.e. at pixel locations (x, y) where $I_x^k(x, y) = I_y^k(x, y) = 0 \ \forall k \in \{1, \cdots, n\}$, we set $\boldsymbol{z_1} = (1, 0)^T$ and $\boldsymbol{z_2} = (0, 1)^T$. It is worth noting that, unlike gray-level images, multi-channel images do not have necessarily level-lines, meaning that the smallest eigenvalues of the structure tensor are not necessarily 0.

We then need to select a set of n vector fields $\boldsymbol{N_1}, \cdots, \boldsymbol{N_n}$ normal to the surface and orthogonal to each other in order to complete the orthonormal moving frame. A natural approach for constructing them is to consider the canonical vectors $\boldsymbol{e_3} = (0, 0, 1, 0, \cdots, 0), \cdots, \boldsymbol{e_{n+2}} = (0, \cdots, 0, 1)$ from which we apply the Gram-Schmidt orthonormalization process to the frame field $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{e_3}, \cdots, \boldsymbol{e_{n+2}})$.

Finally, denoting by P the matrix field encoding the moving frame $(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N_1}, \cdots, \boldsymbol{N_n})$, i.e. the first column of P contains the coordinates of Z_1, the second column the coordinates of Z_2, and the i-th column the coordinates of N_{i-2} for $i \in \{3, \cdots, n+2\}$, the components $(J^1, \cdots, J^{n+2})$ of the $\mathbb{R}^{n+2}$-valued function $(0, 0, I^1, \cdots, I^n)$ in the frame

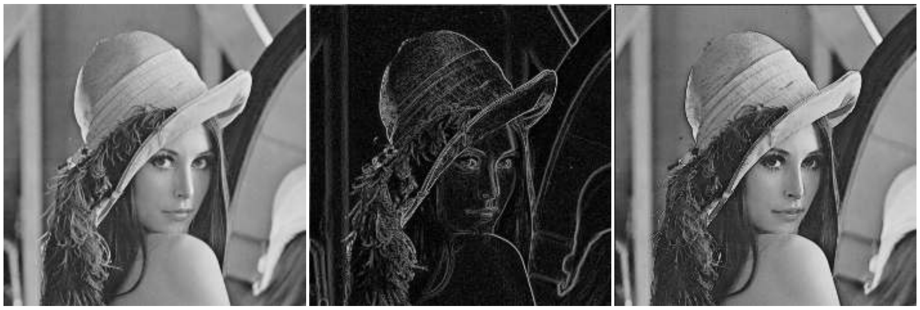

Fig. 2. From left to right: gray-level image "Lena", component J^1, component J^3.

$(\boldsymbol{Z_1}, \boldsymbol{Z_2}, \boldsymbol{N_1}, \cdots, \boldsymbol{N_n})$ are given by

$$\begin{pmatrix} J^1 \\ J^2 \\ J^3 \\ \vdots \\ J^{n+2} \end{pmatrix} = P^{-1} \begin{pmatrix} 0 \\ 0 \\ I^1 \\ \vdots \\ I^n \end{pmatrix} \tag{7}$$

Unlike the gray-level case, it is not possible to plot the moving frame in a trivial way since the surface S lives in a space of dimension greater than or equal to 4.

C. Application to image denoising

The framework we propose for denoising an image while systematically taking into account its local geometry is based on applying image denoising techniques to the components of the image in the moving frame constructed above instead of applying the technique to the image itself. This methodology has already been used in [2],[3],[4] with local regularization/denoising methods, but it can actually be extended to any denoising technique. In this section, we give more details about our approach dealing with gray-level and color images.

1) Gray-level images: In the experiments performed throughout this article, the strategy on gray-level images $I\colon \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}$ is the following:

1) Process I with some denoising technique F and call the output image I_{den}.
2) Compute the components (J^1, J^2, J^3) of I in the moving frame (3), for some scalar μ, with formula (4). Apply the same denoising technique F to the components to obtain the processed components $(J^1_{den}, J^2_{den}, J^3_{den})$. Then, apply the inverse frame change matrix field to the processed components, i.e.
$$\begin{pmatrix} I^1_{denMF} \\ \\ I^2_{denMF} \\ \\ I^3_{denMF} \end{pmatrix} := P \begin{pmatrix} J^1_{den} \\ \\ J^2_{den} \\ \\ J^3_{den} \end{pmatrix} \tag{8}$$
and denote by I_{denMF} the third component I^3_{denMF}.
3) Compare I_{den} and I_{denMF} with the metrics PSNR and SSIM.

Remark 1: We have $\lim_{\mu\to 0} I_{denMF} = I_{den}$ since $\lim_{\mu\to 0}(J^1, J^2, J^3)^T = \lim_{\mu\to 0} P^{-1}(0,0,I)^T = (0,0,I)^T$, in which case $(I^1_{denMF}, I^2_{denMF}, I^3_{denMF})^T = P(J^1_{den}, J^2_{den}, J^3_{den})^T = P(0,0,I_{den})^T = (0,0,I_{den})^T$.

2) Color images: The extension to color images is not straightforward because of the flexibility of the choice of color space and the way in which the moving frame approach can be applied (channel-wise, only to selected channels, or vectorially). We will see in the next two sections that the color space and manner in which the approach is applied both depend on the image denoising technique. However, in all of the experiments performed throughout this article, our approach for color images $I\colon \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ is of the form:

1) Process I with an image denoising technique F and call the output image I_{den}.
2) Apply the same image denoising technique F to the components in some moving frame related to the channels of the image or the full image itself. Then apply the inverse frame change matrix field to the processed components, from which a color image I_{denMV} is reconstructed.
3) Compare I_{den} and I_{denMV} with the metrics PSNR and SSIM. Note that SSIM has been originally designed for gray-level images, and we define the SSIM Index for color images as the mean of the SSIM Index of each color channel.

Finally, we would like to point out that the strategy described above can actually be applied using any moving frame. The problem regarding the choice of the moving frame has already been treated in [4], where we demonstrated numerically that when applying our approach to the Vectorial Total Variation-based denoising method, the best results are obtained with the frame constructed above, but close results are obtained when Z_1, Z_2 are randomly chosen (but still orthornormal) in the tangent planes of the surface parametrized by (1) dealing with gray-level images and (5) for color images. We also observed that very poor results are obtained when Z_1, Z_2 are not in the tangent spaces anymore.

III. The noise level is higher on the intensity values of a gray-level image than on its components in a well-chosen moving frame

The aim of this section is to demonstrate that for properly chosen μ, the components $J^1(I)$ and $J^3(I)$ of a gray-level image I in the moving frame (3), determined by formula (4), are less affected under additive Gaussian noise than the image is. We then deduce that, for a given denoising method, it can be easier to recover the clean components than the clean image directly. In our analysis, we distinguish image edges from the homogeneous regions.

Let $I = a + n$ be a gray-level image resulting from the corruption of an image a with additive Gaussian noise n of zero mean and standard deviation σ.

A. Edges

Inspired by the approach of Bertalmío and Levine [6], we obtain the following result:

Proposition 1: At the locations of the image domain where $|\nabla a| >> |\nabla n|$ (likely the case at contours of I), and assuming that central differences are used to approximate ∇I, we have that for $\mu > 0$,

$$PSNR(J^1(I)) \geq PSNR(I)$$

$$PSNR(J^3(I)) > PSNR(I)$$

Proof: The computation of the local PSNR of the components $J^1(I)$ and $J^3(I)$ requires the knowledge of their amplitude $ampl$. Assuming that I in is the range $[0, 255]$ and central differences are used to compute ∇I, the amplitudes are

$$ampl(J^3(I)) = \quad 255 \tag{9}$$

$$ampl(J^1(I)) = \quad 255 \times \frac{\sqrt{2} \times 127.5\,\mu}{\sqrt{1 + 2(127.5\,\mu)^2}} \tag{10}$$

Indeed, from (3) and (4) we know that

$$J^3(I) = \frac{I}{\sqrt{1 + \mu^2 |\nabla I|^2}} \tag{11}$$

and

$$J^1(I) = \frac{\mu I |\nabla I|}{\sqrt{1 + \mu^2 |\nabla I|^2}} \tag{12}$$

From (11), it is clear that $J^3(I) \in [0, 255]$, meaning that $ampl(J^3(I)) = 255$.
The upper bound of $J^1(I)$ can easily be computed once we notice that $J^1(I)$ is of the form

$$f \colon A \longmapsto I \frac{A}{\sqrt{1 + A^2}}$$

for $A = \mu |\nabla I|$. Then, as the function f is increasing, we deduce that the maximum of $J^1(I)$ is reached when $I = 255$ and $|\nabla I|$ is maximum, this latter quantity depending on the discrete differentiation used. Assuming that central differences are used, we obtain the following upper bound for $J^1(I)$:

$$255 \times \frac{\sqrt{2} \times 127.5\,\mu}{\sqrt{1 + 2(127.5\,\mu)^2}}$$

which corresponds to its amplitude since its lower bound is clearly 0.

At the locations of the image domain where $|\nabla a| >> |\nabla n|$ (likely the case at contours of I), we have

$$\begin{aligned} J^1(I) &= \frac{\mu (a+n) |\nabla I|}{\sqrt{1 + \mu^2 |\nabla I|^2}} \\ &= \frac{\mu a |\nabla a|}{\sqrt{1 + \mu^2 |\nabla a|^2}} \frac{|\nabla I| \sqrt{1 + \mu^2 |\nabla a|^2}}{|\nabla a| \sqrt{1 + \mu^2 |\nabla I|^2}} + \frac{\mu n |\nabla I|}{\sqrt{1 + \mu^2 |\nabla I|^2}} \\ &\approx \frac{\mu a |\nabla a|}{\sqrt{1 + \mu^2 |\nabla a|^2}} + \frac{\mu n |\nabla I|}{\sqrt{1 + \mu^2 |\nabla I|^2}} = J^1(a) + n_1. \end{aligned} \tag{13}$$

Then

$$n_1 = \frac{\mu n |\nabla I|}{\sqrt{1 + \mu^2 |\nabla I|^2}}$$

can be considered as additive noise for the first component $J^1(I)$ with local variance

$$Var(n_1) = Var\left(\frac{\mu n |\nabla I|}{\sqrt{1 + \mu^2 |\nabla I|^2}} \right) = \frac{\mu^2 |\nabla I|^2}{1 + \mu^2 |\nabla I|^2} \sigma^2.$$

Indeed, we can assume $|\nabla I|$ is constant at edge locations and $Var(n) = \sigma^2$ since n is an independent and identically distributed random variable.
Thus at likely edges,

$$PSNR(J^1(I)) = 20 \log_{10} \Bigg(255 \frac{127.5\sqrt{2}\mu}{\sqrt{1 + 2\mu^2 (127.5)^2}} \times \tag{14}$$

$$\frac{\sqrt{1 + \mu^2 |\nabla I|^2}}{\mu |\nabla I| \sigma} \Bigg) \tag{15}$$

$$\geq 20 \log_{10} \left(\frac{255}{\sigma} \right) = PSNR_{loc}(I).$$

As in the computation for $J^1(I)$, at likely edges, $|\nabla I| \approx |\nabla a|$ and thus

$$\begin{aligned} J^3(I) &= \frac{a + n}{\sqrt{1 + \mu^2 |\nabla I|^2}} \\ &= \frac{a}{\sqrt{1 + \mu^2 |\nabla a|^2}} \frac{\sqrt{1 + \mu^2 |\nabla a|^2}}{\sqrt{1 + \mu^2 |\nabla I|^2}} + \frac{n}{\sqrt{1 + \mu^2 |\nabla I|^2}} \\ &\approx \frac{a}{\sqrt{1 + \mu^2 |\nabla a|^2}} + \frac{n}{\sqrt{1 + \mu^2 |\nabla I|^2}} = J^3(a) + n_3 \end{aligned} \tag{16}$$

where

$$n_3 = \frac{n}{\sqrt{1 + \mu^2 |\nabla I|^2}}$$

can be considered as additive noise for the third component $J^3(I)$. Then its local variance at likely edges is approximately

$$Var(n_3) = Var\left(\frac{n}{\sqrt{1 + \mu^2 |\nabla I|^2}} \right) = \frac{\sigma^2}{1 + \mu^2 |\nabla I|^2}.$$

Thus at likely edges, we have that

$$\begin{aligned} PSNR(J^3(I)) &= 20\log_{10}\left(255\frac{\sqrt{1+\mu^2|\nabla I|^2}}{\sigma}\right) \quad (17)\\ &> 20\log_{10}\left(\frac{255}{\sigma}\right) = PSNR(I). \end{aligned}$$

■

According to formula (17), $PSNR(J^3(I))$ is a strictly increasing function of μ that tends to $+\infty$ when $\mu \longrightarrow +\infty$ and to $PSNR(I)$ when $\mu \longrightarrow 0$.

According to formula (14), $PSNR(J^1(I))$ is a decreasing function of μ that tends to $PSNR(I)$ when $\mu \longrightarrow +\infty$ and whose limit at $\mu = 0$ is

$$20log_{10}\left(\frac{255\times 127.5\sqrt{2}}{|\nabla I|\sigma}\right)$$

Hence, we deduce that, at the locations of likely edges, the higher μ is, the better the recovery of the clean component $J^3(a)$, whereas the smaller μ is, the better the recovery of the clean component $J^1(a)$.

Finally, from the fact that $|\nabla I| \approx |\nabla a|$, we have

$$P_{31}(I) \approx P_{31}(a) \qquad P_{33}(I) \approx P_{33}(a) \tag{18}$$

(see (3)) and it follows from the reconstruction equation

$$I_{denMF}: = P_{13}(I)J^1(I)_{den} + P_{33}(I)J^3(I)_{den} \tag{19}$$

(see (8)) that

$$I_{denMF} \approx P_{13}(a)J^1(I)_{den} + P_{33}(a)J^3(I)_{den} \tag{20}$$

From Prop. 1, we intuit that $J^1(I)_{den}$ and $J^3(I)_{den}$ are respectively a better approximation of $J^1(a)$ and $J^3(a)$ than I_{den} is an approximation of a. Hence, from (20) and

$$a = P_{13}(a)J^1(a) + P_{33}(a)J^3(a)$$

we claim that (19) is a better reconstruction of a than I_{den}, from which it follows that the value for the parameter μ that provides the better reconstruction of the edges of the clean image is strictly positive since for $\mu = 0$, we have $I_{denMF} = I_{den}$.

B. Homogeneous regions

We now treat the case where $|\nabla a| << |\nabla n|$ which occurs on homogeneous or slowly varying regions. At such locations in the image domain, we have

$$J^1(I) \approx \frac{\mu I|\nabla n|}{\sqrt{1+\mu^2|\nabla n|^2}} \tag{21}$$

$$J^3(I) \approx \frac{I}{\sqrt{1+\mu^2|\nabla n|^2}} \tag{22}$$

Note that for $\mu > 0$, the range, and thus the fluctuations, of $J^1(I)$ and $J^3(I)$ are attenuated with respect to that of I. In addition, while it is challenging to give a formal proof, the experiments in Table I (explained in the section C. Numerical results) indicate that for small values of $\mu > 0$,

$$PSNR(J^1(I)) > PNSR(I). \tag{23}$$

In the same case, for small values of $\mu > 0$, we obtain the approximation:

$$PNSR(J^3(I)) \approx PNSR(I). \tag{24}$$

Therefore, in homogeneous regions, it also appears that rather than denoising I directly it should be better to denoise $J^1(I)$ and $J^3(I)$.

Furthermore, if $|\nabla a| << |\nabla n|$ then

$$\begin{aligned} I_{denMF} &= P_{13}(I)J^1(I)_{den} + P_{33}(I)J^3(I)_{den}\\ &\approx \frac{\mu|\nabla n|}{\sqrt{1+\mu^2|\nabla n|^2}}J^1(I)_{den} + \frac{1}{\sqrt{1+\mu^2|\nabla n|^2}}J^3(I)_{den}. \end{aligned}$$

so $J^1(I)_{den}$ is given increasingly more weight in the reconstruction as $|\nabla n|$ gets larger. Given the observation in (23) for appropriately chosen μ, this should also benefit the result of the proposed approach. Therefore, in homogeneous regions, we can expect that I_{denMF} should be at least as good as I_{den}, and more likely, better. Experimental results in Section IV corroborate this conjecture.

Finally, from the analysis performed in this section, we conclude that the best reconstruction of the clean image a should be obtained by taking μ very small.

C. Numerical results

From the two previous sections, we know that the parameter μ plays a key role in the quality of the reconstruction of the image. In order to get more insight on the optimal values that μ should take, we run an experiment where we compute the PSNR of the components $J^1(I)$ and $J^3(I)$ for several (constant) values of μ and noise levels σ on the whole Kodak database [16]. Table I reports the results for $\sigma = 5, 10, 15, 20, 25$ and $\mu = 1.0, 0.1, 0.01, 0.005, 0.001, 0.0001$. We observe that the PSNR of the components are higher than the PSNR of the image for $\mu \in]0, 0.005]$ at each of the noise level aforementioned. Note also that the upper bound 0.005 can be raised to 0.01 for the noise levels $\sigma = 5, 10$.

IV. Experiments

Preliminary experiments showed that the value of the parameter μ that provides the best denoising result depends on the image content, the noise level, the denoising method involved, and the measure we use for evaluating the denoised image. As a consequence, it is rather difficult to automatize the value of μ.

Nonetheless, we have found in all our experiments that the value $\mu = 0.001$ systematically provides better results than the value $\mu = 0$, this latter corresponding to the standard denoising method according to Remark 1 in sect. II.C. Experiments also showed that the optimal value is systematically very close to 0.001 when testing non local methods (NLM and BM3D) whatever the image content, the noise level, and the measure we use for evaluating the denoised image are. On the other hand, the optimal value for μ greatly depends on the noise level when testing the local method VTV. In what follows, we report our results for $\mu = 0, \mu = 0.001$ with the three denoising methods VTV, NLM, BM3D, as well as for the optimal values of μ in the VTV case.

TABLE I

AVERAGE VALUES OF THE PSNR FOR THE COMPONENTS J^1, J^3 AND THE IMAGE I OVER THE KODAK DATABASE FOR DIFFERENT NOISE LEVELS AND VALUES OF THE PARAMETER μ.

Noise level	Function	$\mu = 1$	$\mu = 0.1$	$\mu = 0.01$	$\mu = 0.005$	$\mu = 0.001$	$\mu = 0.0001$
$\sigma = 5$	Component J^1	20.51	20.09	34.37	37.84	40.17	40.31
	Component J^3	18.56	26.02	34.24	34.22	34.19	34.19
	Image I	34.19	34.19	34.19	34.19	34.19	34.19
$\sigma = 10$	Component J^1	19.34	15.96	28.21	31.51	33.84	33.97
	Component J^3	16.94	19.84	28.27	28.24	28.21	28.21
	Image I	28.21	28.21	28.21	28.21	28.21	28.21
$\sigma = 15$	Component J^1	18.32	14.16	24.44	27.79	30.09	30.22
	Component J^3	16.32	16.93	24.80	24.77	24.73	24.73
	Image I	24.73	24.73	24.73	24.73	24.73	24.73
$\sigma = 20$	Component J^1	17.37	13.10	21.86	25.12	27.38	27.51
	Component J^3	15.98	15.22	22.38	22.33	22.28	22.27
	Image I	22.27	22.27	22.27	22.27	22.27	22.27
$\sigma = 25$	Component J^1	16.47	12.36	19.89	23.03	25.25	25.38
	Component J^3	15.77	14.10	20.50	20.44	20.37	20.37
	Image I	20.37	20.37	20.37	20.37	20.37	20.37

A. The moving frame approach applied to the Vectorial Total Variation-based denoising method

In this section, we apply our moving frame approach to the Vectorial Total Variation-based denoising method of Bresson and Chan [8], defined as follows on a n-channel image $I_0\colon \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}^n$

$$I_{VTV}\colon = \arg\min_{I} \int_{\Omega} \frac{1}{2\lambda}\|I - I_0\|_2^2 + VTV(I)\ d\Omega \tag{25}$$

where the Vectorial Total Variation $VTV(I)$ of I is defined by

$$\sup\left(\int_{\Omega} \langle I, \nabla^* \eta\rangle_2\, d\Omega; \eta \in C_c^{\infty}(\Omega;\mathbb{R}^n), \|\eta(x)\|_2 \leq 1\, \forall x \in \Omega\right) \tag{26}$$

for ∇^* being the adjoint of the Jacobian operator.
Notice that on gray-level images, this approach reduces to the Rudin-Osher-Fatemi (ROF) model [25].

In what follows, we describe our methodology for both gray-level and color images and report the PSNR and SSIM index values.

1) Gray-level images: As shown by Chambolle [11], the ROF denoising model can be solved through a projection algorithm. In [8], Bresson and Chan showed that their vectorial extension of the ROF denoising model can be solved through a vectorial extension of Chambolle's projection algorithm. Our moving frame approach associated to the ROF denoising model can then be summarized as follows:

1) Take a clean gray-level image a and add Gaussian noise of variance σ to it to create a noisy image I.
2) Apply the ROF denoising model (25) to I using Chambolle's projection algorithm, obtaining a denoised image I_{ROF}.
3) Consider the moving frame associated to I (see (3)), for some parameter value μ, and compute the components $(J^1, J^2, J^3)^T$ of I in this moving frame (see (4)).
   Then, apply the VTV-based denoising model (25) to these components using the vectorial extension of Chambolle's projection algorithm, obtaining the denoised components $(J^1_{VTV}, J^2_{VTV}, J^3_{VTV})$, from which a denoised image $I_{ROFMF}\colon = I^3_{ROFMF}$ is reconstructed using Eq. (8).
4) Compute the PSNR and SSIM index values of I_{ROF} and I_{ROFMF} with respect to the ground truth a.

TABLE II

COMPARISON OF THE STANDARD AND OUR MOVING FRAME APPROACH WITH $\mu = 0.001$ FOR THE VTV-BASED DENOISING METHOD, AT DIFFERENT NOISE LEVELS. AVERAGE PSNR AND SSIM INDEX (X100) OVER THE KODAK DATABASE: THE GRAY-LEVEL CASE.

Approach \ Noise variance	5	10	15	20	25
PSNR Standard	35.44	31.45	29.30	27.83	26.79
PSNR Moving frame	35.54	31.56	29.39	28.02	26.89

Approach \ Noise variance	5	10	15	20	25
SSIM Index Standard	93.73	87.00	81.22	76.05	72.39
SSIM Index Moving frame	93.88	87.22	81.57	77.33	73.24

The ROF model that we compute in step 2 does not have any parameter to tune (assuming that λ in (25) is a Lagrange multiplier associated to the noise level), whereas our moving frame approach associated to that model in step 3 is parametrized by the scalar μ involved in the moving frame.

Table II reports the average PSNR and SSIM index over the Kodak database of both I_{ROF} and I_{ROFMF} images for the parameter value $\mu = 0.001$. We see that our moving frame approach slightly outperforms the standard approach at each noise level tested with respect to both PSNR and SSIM metrics.

Table III reports the average PSNR and SSIM index over the Kodak database for the values of μ that provides the best average PSNR values on the database for our approach. Comparing the results with the ones reported in Table II, we see that the improvement is much more important when optimizing μ on the whole database. Note that optimizing μ for each image would have improved the results reported even more.

TABLE III
OUR MOVING FRAME APPROACH WITH OPTIMAL VALUE OF μ FOR THE VTV-BASED DENOISING METHOD, AT DIFFERENT NOISE LEVELS. AVERAGE PSNR AND SSIM INDEX (X100) OVER THE KODAK DATABASE: THE GRAY-LEVEL CASE.

Approach \ Noise variance	5	10	15	20	25
PSNR Moving frame	36.36	32.23	30.04	28.60	27.49
SSIM Index Moving frame	94.61	88.37	83.22	78.71	74.78

Parameter \ Noise variance	5	10	15	20	25
μ	0.008	0.005	0.005	0.004	0.004

TABLE IV
COMPARISON OF THE STANDARD AND OUR MOVING FRAME APPROACH WITH $\mu = 0.001$ FOR THE VTV-BASED DENOISING METHOD, AT DIFFERENT NOISE LEVELS. AVERAGE PSNR AND SSIM INDEX (X100) OVER THE KODAK DATABASE: THE COLOR CASE.

Approach \ Noise variance	5	10	15	20	25
PSNR Standard	36.31	32.46	30.35	28.79	27.57
PSNR Moving frame	36.45	32.60	30.51	28.96	27.75

Approach \ Noise variance	5	10	15	20	25
SSIM Index Standard	93.82	87.79	82.77	78.24	74.34
SSIM Index Moving frame	93.97	88.10	83.21	78.65	74.98

TABLE V
OUR MOVING FRAME APPROACH WITH OPTIMAL VALUE OF μ FOR THE VTV-BASED DENOISING METHOD, AT DIFFERENT NOISE LEVELS. AVERAGE PSNR AND SSIM INDEX (X100) OVER THE KODAK DATABASE: THE COLOR CASE.

Approach \ Noise variance	5	10	15	20	25
PSNR Moving frame	37.08	33.19	31.05	29.61	28.45
SSIM Index Moving frame	94.66	89.43	84.78	80.89	77.11

Parameter \ Noise variance	5	10	15	20	25
μ	0.0075	0.005	0.0045	0.004	0.004

2) Color images: We proceed as follows.

1) Take a clean color image a and add Gaussian noise of variance σ to it to create the noisy image I.
2) Apply the VTV-based denoising model (25) to I using the vectorial extension of the Chambolle's projection algorithm, obtaining a denoised image I_{VTV}.
3) Consider the moving frame associated to I following the method described in sect. II-B, and compute the components $(J^1, J^2, J^3, J^4, J^5)^T$ of I in this moving frame using (7).
   Then apply the VTV-based denoising model to these components using again the vectorial extension of Chambolle's projection algorithm, obtaining the denoised components $(J^1_{VTV}, J^2_{VTV}, J^3_{VTV}, J^4_{VTV}, J^5_{VTV})^T$, to which the inverse transform of (7) is applied, i.e.
   $$\begin{pmatrix} I^1_{VTVMF} \\ I^2_{VTVMF} \\ I^3_{VTVMF} \\ I^4_{VTVMF} \\ I^5_{VTVMF} \end{pmatrix} : = P \begin{pmatrix} J^1_{VTV} \\ J^2_{VTV} \\ J^3_{VTV} \\ J^4_{VTV} \\ J^5_{VTV} \end{pmatrix}.$$
   The output denoised color image is $I_{VTVMF} \colon = (I^3_{VTVMF}, I^4_{VTVMF}, I^5_{VTVMF})$.
4) Compute the PSNR and SSIM index values of I_{VTV} and I_{VTVMF} with respect to the ground truth a.

As in the gray-level case, the VTV-based denoising model that we compute in step 2 does not have parameters to tune (assuming that λ in (25) is a Lagrange multiplier associated to the noise level), whereas our moving frame approach associated to that model in step 3 is parametrized by the scalar μ determining the moving frame.

Table IV reports the average PSNR and SSIM index over the Kodak database of both I_{VTV} and I_{VTVMF} images for the parameter value $\mu = 0.001$, and Table V reports the average PSNR and SSIM index over the Kodak database for the values of μ that provides the best average PSNR values on the database for our approach. As in the gray-level case, we observe that the improvement is low for $\mu = 0.001$ and much more important when optimizing μ on the whole database. Again, optimizing μ for each image would have improved our results.

In Fig. 3 Rows 1-2, we show an example comparing our approach with the standard VTV-based denoising model. Our method better preserves fine details such as, for example, the texture of the tree leaves and grass.

B. The moving frame approach applied to the Non-Local Means algorithm.

In this section, we apply our framework to the Non-Local Means (NLM) algorithm of Buades et al. [9],[10], defined for a multi-channel image $I = (I^1, \cdots, I^n)$, by

$$I^k_{NLM}(p) \colon = \frac{1}{C(p)} \sum_{q \in \mathcal{B}(p,r)} w(p,q)\, I^k(q) \tag{27}$$

where

$$C(p) = \sum_{q \in \mathcal{B}(p,r)} w(p,q)$$

and $\mathcal{B}(p,r)$ denotes a neighborhood (patch) of the pixel p of size $(2r+1) \times (2r+1)$; The weight function $w(p,q)$ is of the form

$$w(p,q) = e^{\dfrac{-max(d^2 - 2\sigma^2, 0)}{h^2}} \tag{28}$$

where h is a function of the noise level decreasing with the size of the patches, and d is a distance between patches, given by $d^2(\mathcal{B}(p,f), \mathcal{B}(q,f)) =$

$$\frac{1}{n(2f+1)^2} \sum_{i=1}^{n} \sum_{j \in \mathcal{B}(0,f)} (I^i(p+j) - I^i(q+j))^2 \tag{29}$$

In what follows, we describe our approach for both gray-level and color images and report the PSNR and SSIM index values.

1) The case of gray-level images: The way we apply the moving frame approach in the context of the NLM algorithm for gray-level images is described as follows.

1) Take a clean image a and add Gaussian noise of variance σ to it to create the noisy image I.
2) Apply NLM to I using the code available online [10], obtaining a denoised image I_{NLM}.
3) Consider the moving frame associated to I (see (3)), for the parameter value $\mu = 0.001$, and compute the components $(J^1, J^2, J^3)^T$ of I in this moving frame (see (4)).
   Then, apply the NLM algorithm in [10] **channel-wise** to these components, obtaining the denoised components $(J^1_{NLM}, J^2_{NLM}, J^3_{NLM})$, from which a denoised gray-level image $I_{NLM\,MF} : = I^3_{NLM\,MF}$ is constructed (see formula (8)).
4) Compute the PSNR and SSIM index values of I_{NLM} and $I_{NLM\,MF}$ with respect to the ground truth a.

The NLM algorithm is parametrized by the size of the window search (the parameter r in (27)), the size of the patches (the parameter f in (29)), and the function h in (28). In step 2, we test NLM with the default parameters that are given in [10] and which depend on the noise variance σ.

Regarding the parameters used in step 3, let us first point out that, in practice, the component J^2 does not vanish identically, which comes from the fact that the numerical computation of the matrix P^{-1} in (4) does not correspond exactly to the transpose P^T of P. However, since the values of J^2 are negligible, there is no need of denoising that component. Moreover, taking μ small makes the component J^1 have small values too. Hence, J^1 has little weight in the denoising process, meaning that there is no need of optimizing the corresponding parameters, and we can then use the default parameters induced by the noise level σ for denoising the components J^1. Finally, the only parameter of our method is the noise level σ_3 that will determine the parameters r, f, h used for denoising the component J^3.

Table VI reports the average PSNR and SSIM index values over the Kodak database of both I_{NLM} and I_{NLMMF}, as well as the values of the parameter σ_3 that have been used, and that were chosen to give the best PSNR results on average over the entire database for our approach. It is worth noting that, the optimal noise parameter σ_3 is slightly greater than σ in every cases.

We can see that our approach is consistently better, for all noise levels, both in terms of PSNR and SSIM metrics. Note that the biggest improvement occurs when $\sigma = 15$.

TABLE VI
COMPARISON OF THE STANDARD APPROACH AND OUR MOVING FRAME APPROACH WITH $\mu = 0.001$ FOR NLM, AT DIFFERENT NOISE LEVELS. AVERAGE PSNR AND SSIM INDEX (X100), AND OPTIMAL PARAMETER σ_3 OVER THE KODAK DATABASE: THE GRAY-LEVEL CASE.

Approach \ Noise variance	5	10	15	20	25
PSNR Standard	37.41	33.38	31.05	30.04	28.91
PSNR Moving frame	37.52	33.59	31.57	30.12	29.00

Approach \ Noise variance	5	10	15	20	25
SSIM Index Standard	94.96	88.71	82.17	80.34	75.94
SSIM Index Moving frame	95.11	89.54	85.37	81.03	76.95

Parameter \ Noise variance	5	10	15	20	25
σ_3	5.6	11	16	21	26

2) The case of color images: Unlike the VTV-based denoising method in sect. IV.A, the extension to color images is not trivial when dealing with NLM. Indeed, preliminary experiments showed that denoising a color image by applying our previous approach for gray-level images to each of its channels does not improve the results of directly applying NLM to the color image given by [10], regardless of the color space. We tested RGB, Lab, YUV, YC_rC_b, as well as a new color space A_{opp} introduced in Dabov et al. [13]

$$\begin{pmatrix} A^1_{opp} \\ \\ A^2_{opp} \\ \\ A^3_{opp} \end{pmatrix} : = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ \\ 1/2 & 0 & -1/2 \\ \\ 1/4 & -1/2 & 1/4 \end{pmatrix} \begin{pmatrix} R \\ \\ G \\ \\ B \end{pmatrix} \tag{30}$$

where A^1_{opp} encodes the luminance information, and both A^2_{opp} and A^3_{opp} encode the chrominance information. Treating the color image in the RGB color space in a vectorial way by applying a 5D extension of NLM to the components $(J^1, J^2, J^3, J^4, J^5)^T$ following the construction in sect. II.B, as we did for the VTV-based denoising method in sect. IV.A, fails when dealing with NLM. On the other hand, the space proposed in [13] gave the most promising results.

To this end, we were able to improve the standard NLM algorithm by applying our moving frame approach only on the component A^1_{opp} of the color image and taking the components A^2_{opp} and A^3_{opp} from the image denoised directly with NLM. More precisely, our approach is the following one:

1) Take a clean color image a and add Gaussian noise of variance σ to it to create a noisy image I. Compute its component $A^1_{opp}(I)$ in the color space (30).
2) Apply NLM to I using the code available online [10], obtaining a denoised image I_{NLM}, and compute its chrominance components $A^2_{opp}(I_{NLM})$ and $A^3_{opp}(I_{NLM})$ in the color space (30).
3) Apply step 3 of the previous section to the gray-level image $A^1_{opp}(I)$, which produces a denoised image $(A^1_{opp}(I))_{NLMMF}$. Then, consider the image I_{NLMMF} whose components in the color space (30) are $((A^1_{opp}(I))_{NLMMF}, A^2_{opp}(I_{NLM}), A^3_{opp}(I_{NLM}))$.
4) Compute the PSNR and SSIM index values of I_{NLM} and I_{NLMMF} with respect to the ground truth a in the RGB color space.

The NLM algorithm in step 2 was applied with the default parameters for color images, that depend on the noise level σ and which can be found in [10], and the first component J^1 of $A^1_{opp}(I)$ in the moving frame in step 3 was denoised using the default parameters (for gray-level images) induced by the noise level σ. Finally, the only parameter we tune is the noise level σ_3 for denoising the third component J^3.

Table VII reports the average PSNR and SSIM index values

TABLE VII

Comparison of the standard approach and our moving frame approach with $\mu = 0.001$ for NLM, at different noise levels. Average PSNR and SSIM Index (x100), and optimal parameter σ_3 over the Kodak database: the color case.

Approach \ Noise variance	5	10	15	20	25
PSNR Standard	38.35	34.84	32.70	31.18	30.00
PSNR Moving frame	38.76	35.18	33.08	31.61	30.42

Approach \ Noise variance	5	10	15	20	25
SSIM Index Standard	96.01	91.92	87.74	83.74	79.93
SSIM Index Moving frame	96.32	92.69	88.85	85.04	81.08

Parameter \ Noise variance	5	10	15	20	25
σ_3	2.75	6.2	9.6	12.3	16

over the Kodak database of both I_{NLM} and I_{NLMMF}, as well as the values of the parameter σ_3 that have been used, and that were chosen in order to give the best PSNR results on average over the entire database for our approach. As in the gray-level case, our method outperforms the standard approach at each noise level, for both PSNR and SSIM metrics. It is also worth noting that unlike the gray-level case, the optimal value of the noise level σ_3 is systematically lower than σ.

In Fig. 3 Rows 3-4, we illustrate our approach on the image in the database where the improvement of PSNR of our method with respect to the standard approach is one of the highest (0.41dB), and we compare this with the result of the standard NLM algorithm. We can see that our method better preserves fine details like the texture of the ground, for example, which is consistent with the fact that our method is aiming at preserving the local geometry of the processed image.

C. The moving frame approach applied to the Block Matching and 3D Filtering algorithm.

In order to emphasize the consistency of our framework for image denoising, we test it on one of the best algorithms available in the literature: the Block Matching and 3D filtering algorithm BM3D introduced by Dabov et al. [13].

For both gray-level and color images, our approach follows exactly the one of the NLM case, only replacing the NLM algorithm in steps 2 and 3 by the BM3D algorithm whose code is available online [18]. More precisely, in step 2, we test BM3D with the default parameters determined by the noise level σ. In step 3, we apply BM3D on the component J^1 with the default parameters induced by the noise levels $\sigma_1 = \sigma$, and we test several values of the noise levels σ_3 when applying BM3D on the component J^3, in order to find the ones that provide the best average results on the whole database in terms of PSNR.

TABLE VIII

Comparison of the standard approach and our moving frame approach with $\mu = 0.001$ for $BM3D$, at different noise levels. Average PSNR and SSIM Index (x100), and optimal parameter σ_3 over the Kodak database: the gray-level case.

Approach \ Noise variance	5	10	15	20	25
PSNR Standard	38.23	34.34	32.26	30.89	29.88
PSNR Moving frame	38.25	34.38	32.31	30.93	29.92

Approach \ Noise variance	5	10	15	20	25
SSIM Index Standard	95.71	91.38	87.52	84.19	81.32
SSIM Index Moving frame	95.74	91.49	87.71	84.38	81.44

Parameter \ Noise variance	5	10	15	20	25
σ_3	4.9	9.7	14.4	19.1	23.9

TABLE IX

Comparison of the standard approach and our moving frame approach with $\mu = 0.001$ for $BM3D$, at different noise levels. Average PSNR and SSIM Index (x100), and optimal parameter σ_3 over the Kodak database: the color case.

Approach \ Noise variance	5	10	15	20	25
PSNR Standard	40.35	36.50	34.32	32.83	31.72
PSNR Moving frame	40.38	36.53	34.36	32.88	31.77

Approach \ Noise variance	5	10	15	20	25
SSIM Index Standard	97.05	94.22	91.49	88.89	86.50
SSIM Index Moving frame	97.08	94.26	91.62	89.06	86.71

Parameter \ Noise variance	5	10	15	20	25
σ_3	2.75	5.6	8.2	11.1	13.8

Results on gray-level images are reported in table VIII and results on color images are reported in table IX. In both cases, we can see that our approach is consistently better, though very slightly, for all noise levels, and for both PSNR and SSIM metrics. The increase in PSNR that we obtain, while modest, is in agreement with the optimality bounds estimated by Levin and Nadler [20], and Chaterjee and Milanfar [12], and are comparable with the ones obtained through recent boosting techniques [24]. Finally, we notice that the optimal values of σ_3 are systematically lower than the true noise level σ.

In Fig. 3 Rows 5-6, we show an example of our approach where we compare the result with the one of the standard BM3D denoising model. We can see that our method preserves better fine details like the wooden decoration engraved in the balcony, for example.

V. Conclusion

In this paper, we have developed a framework that enables any denoising method to take more into account the local geometry of the image to be denoised by preserving the moving frame describing the graph of a scaled version of the image. Experiments with the VTV-based denoising method, NLM and BM3D algorithms on both gray-level and color images tested over the Kodak database showed that our strategy systematically improves the denoising method it is applied to, in terms of PSNR and SSIM metrics. The fact that we have been able to improve the performance of three denoising methods of different types: a local variational method, a patch-based method, and a method combining a patch-based approach with a filtering in spectral domain approach, demonstrates the consistency of our methodology.

In the proposed strategy for denoising, we either combine the components into a single vector-valued function to which we apply a denoising method (VTV) or treat them separately applying the same denoising method but with different

parameters (NLM and BM3D). However, as the components have different geometric meaning, one shall wonder whether they should not rather be denoised with different denoising methods, and we are currently investigating that point.
Further work will be devoted to applying our framework to denoising methods that treat images whose noise model is unknown; this setting is more realistic, and there is more room for improvement than with methods treating additive Gaussian noise.

Acknowledgment

This work was supported by the European Research Council, Starting Grant ref. 306337, by the Spanish government, grant ref. TIN2012-38112, and by the Icrea Academia Award. The last author is supported in part by NSF-DMS 1320829.

References

[1] S.P. Awate and R.T. Whitaker,“Higher-order image statistics for unsupervised, information-theoretic, adaptive, image filtering”, *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 44-51, 2005.

[2] T. Batard and M. Berthier,“Spinor Fourier transform for image processing”, *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 4, pp. 605-613, 2013.

[3] T. Batard and M. Bertalmío,“Generalized gradient on vector bundle-application to image denoising”, *Lecture Notes Comput. Sci.*, vol. 7893, pp. 12-23, 2013.

[4] T. Batard and M. Bertalmío, “On covariant derivatives and their applications to image regularization ”, *SIAM J. Imag. Sci.*, vol. 7, no. 4, pp. 2393-2422, 2014.

[5] M. Bertalmío, *Image Processing for Cinema*, Boca Raton, FL: CRC Press; Taylor and Francis, 2014.

[6] M. Bertalmío and S. Levine,“Denoising an image by denoising its curvature image ”, *SIAM J. Imag. Sci.*, vol. 7, no. 2, pp. 187-201, 2014.

[7] 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, 1998.

[8] X. Bresson and T.F. Chan, “Fast dual minimization of the vectorial total variation norm and applications to color image processing”, *Inverse Probl. Imaging*, vol. 2, no. 4, pp. 455-484, 2008.

[9] A. Buades, B. Coll and J.-M. Morel, “A non-local algorithm for image denoising”, *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 60-65, 2005.

[10] A. Buades, B. Coll and J.-M. Morel,“Non-local means denoising”, *Image Processing On Line*, vol. 1, 2011.

[11] A. Chambolle, “An algorithm for total variation minimization and applications”, *J. Math. Imaging Vis.*, vol. 20, pp. 89-97, 2004.

[12] P. Chaterjee and P. Milanfar,“Is denoising dead?”, *IEEE Trans. Image Process*, vol. 19, no. 4, pp. 895-911, 2010.

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

[14] G. Ghimpeţeanu, T. Batard, M. Bertalmío and S. Levine, “Denoising an image by denoising its components in a moving frame”, *Lecture Notes Comput. Sci.*, vol. 8509, pp. 375-383, 2014.

[15] J. Hahn, X.-C. Tai, S. Borok and A. M. Bruckstein, “Orientation-matching minimization for image denoising and inpainting”, *Int. J. Comput. Vis.*, vol. 92, no. 3, pp. 308-324, 2011.

[16] http://r0k.us/graphics/kodak/

[17] M. Lebrun, M. Colom, A. Buades and J.M. Morel,“Secrets of image denoising cuisine”, *Acta Numerica*, vol. 21, no. 1, pp. 475-576, 2012.

[18] M. Lebrun, “An analysis and implementation of the BM3D image denoising method”, *Image Processing On Line*, vol. 2, pp.175-213, 2012.

[19] M. Lebrun, M. Colom and J.M. Morel,“The noise clinic: a universal blind denoising algorithm”, *Proc. IEEE Int. Conf. Image Process.*, 2014.

[20] A. Levin and B. Nadler, “Natural image denoising: Optimality and inherent bounds”, *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 2833-2840, 2011.

[21] M. Lysaker, S. Osher and X.C. Tai,“Noise removal using smoothed normals and surface fitting”, *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1345-1357, 2004.

[22] S. Osher, M. Burger, D. Goldfarb, J. Xu and W. Yin, “An iterative regularization method for total variation-based image restoration”, *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp.460

[23] T. Rahman, X.-C. Tai, and S. Osher, “A tv-stokes denoising algorithm”, *Lecture Notes Comput. Sci.*, vol. 4485, pp. 473-483, 2007. vol. 8509, pp. 375-383, 2014.

[24] Y. Romano, M. Elad,“Boosting of image denoising techniques”, *SIAM J. Imag. Sci.*, vol. 8, no.2, pp. 1187-1219, 2015.

[25] L.I. Rudin, S. Osher and E. Fatemi,“Nonlinear total variation based noise removal algorithms”, *Physica D: Nonlinear Phenomena.*, vol. 60, no. 1-4, pp. 259-268, 1992.

[26] Z. Wang and A.C. Bovik,“A universal image quality index”, *IEEE Signal Process. Lett.*, vol. 9, pp. 81-84, 2002.

[27] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli,“Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600-612, 2004.

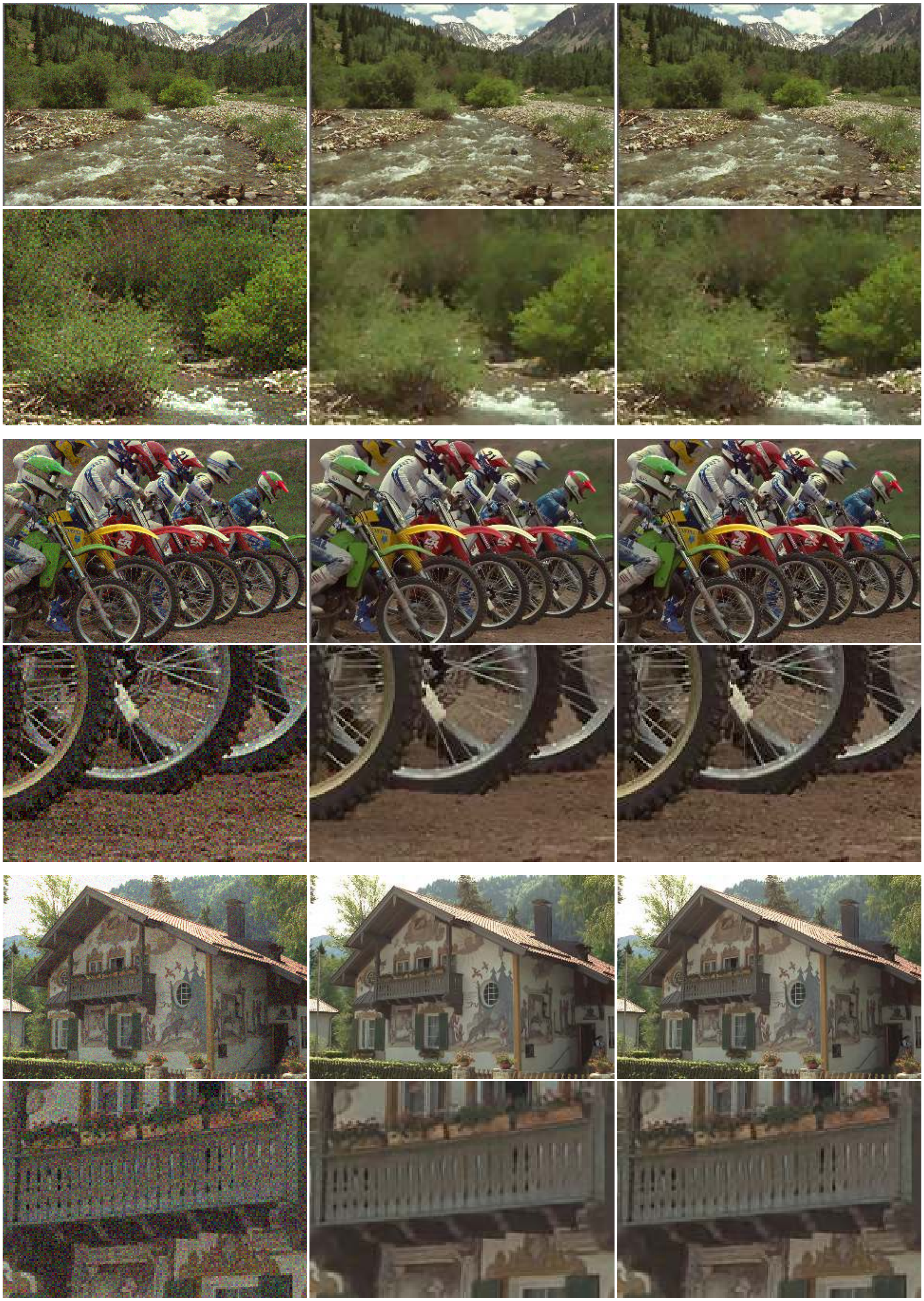

Fig. 3. Examples of our framework for different denoising algorithms. Row 1: our framework with VTV [8] as denoising method. Left: noisy image “kodim13” with $\sigma = 15$. Middle: result of applying VTV to the image, PSNR=27.00. Right: our result, applying VTV to the components, PSNR=27.56. Row 2: zoomed-in details from the first row. Row 3: our framework with NLM [9] as denoising method. Left: noisy image “kodim05” with $\sigma = 20$. Middle: result of applying NLM to the image, PSNR=29.29. Right: our result, applying NLM to the components, PSNR=29.70. Row 4: zoomed-in details from the third row. Row 5: our framework with BM3D [13] as denoising method. Left: noisy image “kodim24” with $\sigma = 20$. Middle: result of applying BM3D to the image, PSNR=31.26. Right: applying BM3D to the components with optimized parameters, PSNR=31.40. Row 6: zoomed-in details from the fifth row.