A Decomposition Framework for Image Denoising Algorithms
Gabriela Ghimpeteanu, 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 et 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 Bertalmio [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 Bertalmio 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
 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 technique that is applied. We have been able to improve

\[ \psi(x, y) \rightarrow (x, y, \mu I(x, y)) \]  

(1)

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

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

\[ Z_i = \frac{d\psi(z_1)}{\|d\psi(z_1)\|_2}, \quad i = 1, 2 \]  

(2)

where \(d\psi\) stands for the differential of \(\psi\), which maps vector fields on \(\Omega\) to tangent vector fields of \(S\). The expression of the unit normal \(N\) is then obtained as the vectorial product between \(Z_1\) and \(Z_2\).

The explicit expressions of the vector fields \(Z_1, Z_2, N\) are given by the matrix field

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

(3)

where the coordinates of the vector field \(Z_1\) are given in the first column, the coordinates of \(Z_2\) in the second column, and the coordinates of \(N\) in the third column.

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

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

II. IMAGE DECOMPOSITION IN A MOVING FRAME

A. The gray-level case

Let \(I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}\) 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 \(\nabla I\) the gradient of \(I\).

Our image decomposition model for \(I\) is a two-stages approach: first, we construct an orthonormal moving frame \((Z_1, Z_2, N)\) of \((\mathbb{R}^3, ||\cdot||_2)\) over \(\Omega\) 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 \(\mu I\) of \(I\), for \(\mu \in [0, 1]\), and its graph, which is the surface \(S\) in \(\mathbb{R}^3\) parametrized by

\[ \psi(x, y) \rightarrow (x, y, \mu I(x, y)) \]  

(1)

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

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

\[ Z_i = \frac{d\psi(z_1)}{\|d\psi(z_1)\|_2}, \quad i = 1, 2 \]  

(2)

where \(d\psi\) stands for the differential of \(\psi\), which maps vector fields on \(\Omega\) to tangent vector fields of \(S\). The expression of the unit normal \(N\) is then obtained as the vectorial product between \(Z_1\) and \(Z_2\).

The explicit expressions of the vector fields \(Z_1, Z_2, N\) are given by the matrix field

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

(3)

where the coordinates of the vector field \(Z_1\) are given in the first column, the coordinates of \(Z_2\) in the second column, and the coordinates of \(N\) in the third column.

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

Denoting by \((e_1, e_2, e_3)\) the orthonormal frame of \((\mathbb{R}^3, ||\cdot||_2)\), where \(e_1 = (1, 0, 0), e_2 = (0, 1, 0), e_3 = (0, 0, 1)\), the matrix \(P\) in (3) is nothing but the frame change field from
which is the surface components encode the local geometry of \( I(0, 0, I) \) in the new frame, denoted by \((J^1, J^2, J^3)\), are given by

\[
\begin{pmatrix}
  J^1 \\
  J^2 \\
  J^3
\end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix} P^{-1}
\] (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 \( Z_1 \) 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 \( \mu \) 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 \( \mu \) 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 \( I = (I^1, \cdots, I^n) : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^n, n > 1 \), by following a similar approach: first, we construct an orthonormal moving frame \((Z_1, Z_2, N_1, \cdots, N_n)\) of \((\mathbb{R}^{n+2}, \| \cdot \|_2)\) over \( \Omega \) 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 \( \mu I \) of \( I \), for \( \mu \in [0, 1] \), and its graph, which is the surface \( S \) in \( \mathbb{R}^{n+2} \) parametrized by

\[
\psi : (x, y) \rightarrow (x, y, \mu I^1(x, y), \cdots, \mu I^n(x, y))
\] (5)

The moving frame \((Z_1, Z_2, N_1, \cdots, N_n)\) we consider is then the following: the vector field \( Z_1 \) is tangent to the surface \( S \) and indicates the direction of the steepest slope at each point of \( S \); the vector field \( Z_2 \) is tangent to \( S \) and indicates the direction of the lowest slope at each point of \( S \), and \( N_1, \cdots, N_n \) 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 \((Z_1, Z_2, N_1, \cdots, N_n)\) can be constructed as follows. As in the gray-level case, \( Z_1 \) and \( Z_2 \) can be recovered from the directions \( z_1 \) and \( z_2 \) of highest and lowest variations of the scaled image \( \mu I \) under the map (2), these latter being the eigenvectors of the structure tensor associated to \( \mu I \), which is given by

\[
\begin{pmatrix}
  \sum_{k=1}^n (\mu I_k^1)^2 & \sum_{k=1}^n \mu I_k^1 I_k^2 \\
  \sum_{k=1}^n \mu I_k^1 I_k^2 & \sum_{k=1}^n (\mu I_k^2)^2
\end{pmatrix}
\] (6)

On homogeneous regions, i.e. at pixel locations \((x, y)\) where \( I_k^2(x, y) = I_k^1(x, y) = 0 \, \forall k \in \{1, \cdots, n\} \), we set \( z_1 = (1, 0)^T \) and \( z_2 = (0, 1)^T \). 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 \( N_1, \cdots, N_n \) 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 \( e_2 = (0, 0, 1, 0, \cdots, 0) \), \( e_{n+2} = (0, \cdots, 0, 1) \) from which we apply the Gram-Schmidt orthonormalization process to the frame field \((Z_1, Z_2, e_3, \cdots, e_{n+2})\).

Finally, denoting by \( P \) the matrix field encoding the moving frame \((Z_1, Z_2, N_1, \cdots, N_n)\), 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 \( N_i \) 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...
(Z_1, Z_2, N_1, \ldots, N_n) are given by
\[
\begin{pmatrix}
J^1 \\
J^2 \\
J^3 \\
\vdots \\
J^{n+2}
\end{pmatrix} = P^{-1}
\begin{pmatrix}
0 \\
0 \\
I \\
\vdots \\
I^n
\end{pmatrix}
\] (7)

Remark 1: We have \( \lim_{\mu \to 0} I_{\text{denMF}} = I_{\text{den}} \) 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 \((I_{\text{denMF}}, I_{\text{denMF}}^2, I_{\text{denMF}}^3)^T = P(J_{\text{den}}, J_{\text{den}}, J_{\text{den}})^T = P(0, 0, I_{\text{den}})^T = (0, 0, I_{\text{den}})^T \).

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 \( I: \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3 \) is of the form:

1) Process \( I \) with an image denoising technique \( F \) and call the output image \( I_{\text{den}} \).
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 \( I_{\text{denMV}} \) is reconstructed.
3) Compare \( I_{\text{den}} \) and \( I_{\text{denMV}} \) 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 orthonormal) 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 \( \mu \), 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 \( \sigma \).

A. Edges

Inspired by the approach of Bertalmio 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 \( \nabla I \), 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 amplitudes \( \text{ampl} \). Assuming that \( I \) is in the range \([0,255]\) and central differences are used to compute \( \nabla I \), the amplitudes are

\[
\text{ampl}(J^3(I)) = \frac{255}{\sqrt{1 + 2(127.5\mu)^2}} \tag{9}
\]

\[
\text{ampl}(J^1(I)) = \frac{255}{\sqrt{1 + 2(127.5\mu)^2}} \tag{10}
\]

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

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

and

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

From (11), it is clear that \( J^3(I) \in [0,255] \), meaning that \( \text{ampl}(J^3(I)) = 255 \).

The upper bound of \( J^1(I) \) can easily be computed once we notice that \( J^1(I) \) is of the form

\[
f: A \mapsto I \mapsto \frac{A}{\sqrt{1 + A^2}}
\]

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 \( J^1(I) \):

\[
255 \times \sqrt{\frac{\cdot 127.5\mu}{1 + 2(127.5\mu)^2}}
\]

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

\[
J^1(I) = \frac{\mu(a + n)|\nabla I|}{\sqrt{1 + \mu^2|\nabla I|^2}}
\]

\[
= \frac{\mu a|\nabla a|}{\sqrt{1 + \mu^2|\nabla a|^2}} \cdot \frac{|\nabla I|}{\sqrt{1 + \mu^2|\nabla I|^2}} + \frac{\mu n|\nabla I|}{\sqrt{1 + \mu^2|\nabla I|^2}}
\]

\[
\approx \frac{\mu a|\nabla a|}{\sqrt{1 + \mu^2|\nabla a|^2}} + \frac{\mu n|\nabla I|}{\sqrt{1 + \mu^2|\nabla I|^2}} = J^1(a) + n_1.
\]

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

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

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

\[
PSNR(J^1(I)) = 20\log_{10}\left( \frac{255}{\sqrt{1 + 2(127.5\mu)^2}} \times \frac{\sqrt{1 + \mu^2|\nabla I|^2}}{\mu|\nabla I|} \right) \tag{14}
\]

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

As in the computation for \( J^3(I) \), at likely edges, \( |\nabla I| \approx |\nabla a| \) and thus

\[
J^3(I) = \frac{a + n}{\sqrt{1 + \mu^2|\nabla I|^2}}
\]

\[
= \frac{a}{\sqrt{1 + \mu^2|\nabla a|^2}} \cdot \frac{1 + \mu^2|\nabla a|^2}{\sqrt{1 + \mu^2|\nabla I|^2}} + \frac{n}{\sqrt{1 + \mu^2|\nabla I|^2}}
\]

\[
\approx \frac{a}{\sqrt{1 + \mu^2|\nabla a|^2}} + \frac{n}{\sqrt{1 + \mu^2|\nabla I|^2}} = J^3(a) + n_3
\]

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

\[
\text{Var}(n_3) = \text{Var}\left( \frac{n}{\sqrt{1 + \mu^2|\nabla I|^2}} \right) = \frac{\sigma^2}{1 + \mu^2|\nabla I|^2}.
\]
Thus at likely edges, we have that

\[
PSNR(J^3(I)) = 20\log_{10} \left( \frac{255 \sqrt{1 + \mu^2 |\nabla I|^2}}{\sigma} \right) \quad (17)
\]

\[
> 20\log_{10} \left( \frac{255}{\sigma} \right) = PSNR(I).
\]

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

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

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

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

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

\[
P_{31}(I) \approx P_{31}(a) \quad P_{33}(I) \approx P_{33}(a)
\]

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

\[
I_{denMF}^*: = P_{33}(I)J^1(I)_{den} + P_{33}(a)J^3(I)_{den}
\]

(see (8)) that

\[
I_{denMF} \approx P_{33}(a)J^1(I)_{den} + P_{33}(a)J^3(I)_{den}
\]

(19)

(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

\[
a = P_{33}(a)J^1(a) + P_{33}(a)J^3(a)
\]

we claim that (19) is a better reconstruction of \(a\) than \(I_{den}\), from which it follows that the value for the parameter \(\mu\) 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 \(|\nabla a| \ll |\nabla n|\) which occurs on homogeneous or slowly varying regions. At such locations in the image domain, we have

\[
J^1(I) \approx \frac{\mu |\nabla n|}{\sqrt{1 + \mu^2 |\nabla n|^2}}
\]

(21)

\[
J^3(I) \approx \frac{I}{\sqrt{1 + \mu^2 |\nabla n|^2}}
\]

(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 \(\mu > 0\),

\[
PSNR(J^1(I)) > PNSR(I).
\]

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

\[
PSNR(J^3(I)) \approx PNSR(I).
\]

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 \(|\nabla a| \ll |\nabla n|\) then

\[
I_{denMF} = P_{33}(I)J^1(I)_{den} + P_{33}(a)J^3(I)_{den}
\]

\[
\approx \frac{\mu |\nabla n|}{\sqrt{1 + \mu^2 |\nabla n|^2}}J^1(I)_{den} + \frac{1}{\sqrt{1 + \mu^2 |\nabla n|^2}}J^3(I)_{den},
\]

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 \(\mu\), 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 \(\mu\) very small.

**C. Numerical results**

From the two previous sections, we know that the parameter \(\mu\) plays a key role in the quality of the reconstruction of the image. In order to get more insight on the optimal values that \(\mu\) 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 \(\mu\) and noise levels \(\sigma\) 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 \(\mu\) 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 \(\mu\).

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 \(\mu\) 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 \(\mu\) in the VTV case.
TABLE I

<table>
<thead>
<tr>
<th>Noise level</th>
<th>Function</th>
<th>$\mu = 1$</th>
<th>$\mu = 0.1$</th>
<th>$\mu = 0.01$</th>
<th>$\mu = 0.005$</th>
<th>$\mu = 0.001$</th>
<th>$\mu = 0.0001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma = 5$</td>
<td>Component $J^1$</td>
<td>20.51</td>
<td>20.09</td>
<td>34.37</td>
<td>37.84</td>
<td>40.17</td>
<td>40.71</td>
</tr>
<tr>
<td></td>
<td>Component $J^3$</td>
<td>18.56</td>
<td>26.02</td>
<td>34.24</td>
<td>34.22</td>
<td>34.19</td>
<td>34.19</td>
</tr>
<tr>
<td></td>
<td>Image $I$</td>
<td>34.19</td>
<td>34.19</td>
<td>34.19</td>
<td>34.19</td>
<td>34.19</td>
<td>34.19</td>
</tr>
<tr>
<td>$\sigma = 10$</td>
<td>Component $J^1$</td>
<td>19.34</td>
<td>15.96</td>
<td>28.21</td>
<td>31.51</td>
<td>33.84</td>
<td>33.97</td>
</tr>
<tr>
<td></td>
<td>Component $J^3$</td>
<td>16.94</td>
<td>19.84</td>
<td>28.27</td>
<td>28.24</td>
<td>28.21</td>
<td>28.21</td>
</tr>
<tr>
<td></td>
<td>Image $I$</td>
<td>28.21</td>
<td>28.21</td>
<td>28.21</td>
<td>28.21</td>
<td>28.21</td>
<td>28.21</td>
</tr>
<tr>
<td>$\sigma = 15$</td>
<td>Component $J^1$</td>
<td>18.32</td>
<td>14.16</td>
<td>24.44</td>
<td>27.79</td>
<td>30.09</td>
<td>30.32</td>
</tr>
<tr>
<td></td>
<td>Component $J^3$</td>
<td>16.32</td>
<td>16.93</td>
<td>24.80</td>
<td>24.77</td>
<td>24.73</td>
<td>24.73</td>
</tr>
<tr>
<td></td>
<td>Image $I$</td>
<td>24.73</td>
<td>24.73</td>
<td>24.73</td>
<td>24.73</td>
<td>24.73</td>
<td>24.73</td>
</tr>
<tr>
<td>$\sigma = 20$</td>
<td>Component $J^1$</td>
<td>17.37</td>
<td>13.10</td>
<td>21.86</td>
<td>25.12</td>
<td>27.38</td>
<td>27.51</td>
</tr>
<tr>
<td></td>
<td>Component $J^3$</td>
<td>15.98</td>
<td>15.22</td>
<td>22.38</td>
<td>23.33</td>
<td>22.28</td>
<td>22.27</td>
</tr>
<tr>
<td></td>
<td>Image $I$</td>
<td>22.27</td>
<td>22.27</td>
<td>22.27</td>
<td>22.27</td>
<td>22.27</td>
<td>22.27</td>
</tr>
<tr>
<td>$\sigma = 25$</td>
<td>Component $J^1$</td>
<td>16.47</td>
<td>12.36</td>
<td>19.89</td>
<td>23.03</td>
<td>25.25</td>
<td>25.38</td>
</tr>
<tr>
<td></td>
<td>Component $J^3$</td>
<td>15.77</td>
<td>14.10</td>
<td>20.50</td>
<td>20.44</td>
<td>20.37</td>
<td>20.37</td>
</tr>
</tbody>
</table>

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 $I_0: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^n$

$$I_{VT}^V: = \arg \min_I \int_{\Omega} \frac{1}{2\lambda} ||I - I_0||_2^2 + VTV(I) \, d\Omega$$

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

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

for $\nabla^*$ 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 $\sigma$ 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 $\mu$, 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 $(J_{1VTV}, J_{2VTV}, J_{3VTV})$, from which a denoised image $I_{ROFMF} = J_{3ROFMF}$ 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 $a$.

The ROF model that we compute in step 2 does not have any parameter to tune (assuming that $\mu$ is parametrized by the scalar $\mu$ involved in the moving frame).

The ROF model that we compute in step 2 does not have any parameter to tune (assuming that $\lambda$ 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 $\mu$ 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 $\mu$ 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 $\mu$ on the whole database. Note that optimizing $\mu$ for each image would have improved the results reported even more.
2) Color images: We proceed as follows.

1) Take a clean color image \( a \) and add Gaussian noise of variance \( \sigma \) 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 \( (J_1^1, J_2^1, J_3^1, J_4^1, J_5^1)^T \) 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 \( I_{VTvMF} = (I_1^{VTvMF}, I_2^{VTvMF}, I_3^{VTvMF}, I_4^{VTvMF}, I_5^{VTvMF}) \).

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 \( \lambda \) 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 \( \mu \) 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 \( \mu \) 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 \( \mu \) on the whole database. Again, optimizing \( \mu \) 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

\[
I_{NLM}^k(p) = \frac{1}{C(p)} \sum_{q \in B(p,r)} w(p, q) I^k(q) \tag{27}
\]

where

\[
C(p) = \sum_{q \in B(p,r)} w(p, q)
\]

and \( B(p, r) \) 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

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

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 \( d^2(B(p, f), B(q, f)) = \frac{1}{n(2f+1)^2} \sum_{i=1}^{n} \sum_{j \in B(f, j)} (I^i(p + j) - I^i(q + j))^2 \) \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 \( \sigma \) to it to create the noisy image \( I \).
2) Apply NLM to \( I \) using the code available online [10], obtaining a denoised image \( I_{\text{NLM}} \).
3) Consider the moving frame associated to \( I \) (see (3)), for the parameter value \( \mu = 0.001 \), and which depend on the noise variance \( \sigma \). Then, apply the NLM algorithm in [10] channel-wise to these components, obtaining the denoised components \( (J_{\text{NLM}}^1, J_{\text{NLM}}^2, J_{\text{NLM}}^3)^T \) of \( I \) in this moving frame (see (4)).
4) Compute the PSNR and SSIM index values of \( I_{\text{NLM}} \) and \( I_{\text{NLM MF}} \) 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 \( \sigma \). 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 \( \mu \) 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 \( \sigma \) for denoising the components \( J^1 \). Finally, the only parameter of our method is the noise level \( \sigma_3 \) 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 \( I_{\text{NLM}} \) and \( I_{\text{NLM MF}} \), as well as the values of the parameter \( \sigma_3 \) 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 \( \sigma_3 \) is slightly greater than \( \sigma \) 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 \( \sigma = 15 \).

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 \( A_{\text{opp}} \) introduced in Dabov et al. [13]

\[
\begin{align*}
A_{\text{opp}}^1 & = \begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \\
A_{\text{opp}}^2 & = \begin{pmatrix} 1/2 & 0 & -1/2 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \\
A_{\text{opp}}^3 & = \begin{pmatrix} 1/4 & -1/2 & 1/4 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}
\end{align*}
\]

where \( A_{\text{opp}}^1 \) encodes the luminance information, and both \( A_{\text{opp}}^2 \) and \( A_{\text{opp}}^3 \) 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 \( A_{\text{opp}}^1 \) of the color image and taking the components \( A_{\text{opp}}^2 \) and \( A_{\text{opp}}^3 \) 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 \( \sigma \) to it to create a noisy image \( I \). Compute its chrominance components \( A_{\text{opp}}^2(I_{\text{NLM}}) \) and \( A_{\text{opp}}^3(I_{\text{NLM}}) \) in the color space (30).
2) Apply NLM to \( I \) using the code available online [10], obtaining a denoised image \( I_{\text{NLM}} \), and compute its chrominance components \( A_{\text{opp}}^2(I_{\text{NLM}}) \) and \( A_{\text{opp}}^3(I_{\text{NLM}}) \) in the color space (30).
3) Apply step 3 of the previous section to the gray-level image \( A_{\text{opp}}^1(I_{\text{NLM}}) \), which produces a denoised image \( A_{\text{opp}}^1(I_{\text{NLM MF}}) \). Then, consider the image \( I_{\text{NLM MF}} \) whose components in the color space (30) are \( (A_{\text{opp}}^1(I_{\text{NLM MF}}), A_{\text{opp}}^2(I_{\text{NLM}}), A_{\text{opp}}^3(I_{\text{NLM}})) \).
4) Compute the PSNR and SSIM index values of \( I_{\text{NLM}} \) and \( I_{\text{NLM MF}} \) 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 \( \sigma \) and which can be found in [10], and the first component \( J^1 \) of \( A_{\text{opp}}^1(I) \) in the moving frame in step 3 was denoised using the default parameters (for gray-level images) induced by the noise level \( \sigma \). Finally, the only parameter we tune is the noise level \( \sigma_3 \) for denoising the third component \( J^3 \).

Table VII reports the average PSNR and SSIM index values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Noise variance</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma_3 )</td>
<td>5.6</td>
<td>11</td>
<td>16</td>
<td>21</td>
<td>26</td>
<td></td>
</tr>
</tbody>
</table>

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 \( \sigma \) to it to create the noisy image \( I \).
2) Apply NLM to \( I \) using the code available online [10], obtaining a denoised image \( I_{\text{NLM}} \).
3) Consider the moving frame associated to \( I \) (see (3)), for the parameter value \( \mu = 0.001 \).
4) Compute the PSNR and SSIM index values of \( I_{\text{NLM}} \) and \( I_{\text{NLM MF}} \) with respect to the ground truth \( a \).
over the Kodak database of both $I_{\text{NLM}}$ and $I_{\text{NLM, MF}}$, as well as the values of the parameter $\sigma_3$ 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 $\sigma_3$ is systematically lower than $\sigma$.

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 $\sigma$. In step 3, we apply BM3D on the component $J^3$ with the default parameters induced by the noise levels $\sigma_1 = \sigma$, and we test several values of the noise levels $\sigma_3$ 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.

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 $\sigma_3$ are systematically lower than the true noise level $\sigma$.

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

---

**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 $\sigma_3$ over the Kodak database: the color case.

<table>
<thead>
<tr>
<th>Approach \ Noise variance</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR Standard</td>
<td>38.35</td>
<td>34.84</td>
<td>32.70</td>
<td>31.18</td>
<td>30.00</td>
</tr>
<tr>
<td>PSNR Moving frame</td>
<td>38.76</td>
<td>35.18</td>
<td>33.08</td>
<td>31.61</td>
<td>30.42</td>
</tr>
</tbody>
</table>

---

**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 $\sigma_3$ over the Kodak database: the gray-level case.

<table>
<thead>
<tr>
<th>Approach \ Noise variance</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR Standard</td>
<td>38.23</td>
<td>34.34</td>
<td>32.36</td>
<td>30.89</td>
<td>29.88</td>
</tr>
<tr>
<td>PSNR Moving frame</td>
<td>38.25</td>
<td>34.38</td>
<td>32.31</td>
<td>30.93</td>
<td>29.92</td>
</tr>
</tbody>
</table>

---

**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 $\sigma_3$ over the Kodak database: the color case.

<table>
<thead>
<tr>
<th>Approach \ Noise variance</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR Standard</td>
<td>40.35</td>
<td>36.50</td>
<td>34.52</td>
<td>32.83</td>
<td>31.72</td>
</tr>
<tr>
<td>PSNR Moving frame</td>
<td>40.38</td>
<td>36.53</td>
<td>34.56</td>
<td>32.88</td>
<td>31.77</td>
</tr>
</tbody>
</table>

---

**Parameter \ Noise variance**

<table>
<thead>
<tr>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_3$</td>
<td>2.75</td>
<td>5.6</td>
<td>8.2</td>
<td>11.3</td>
</tr>
</tbody>
</table>
parameters (NLМ 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


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


