



# Speckle Reduction in Matrix-Log Domain for Synthetic Aperture Radar Imaging

Charles-Alban Deledalle<sup>1</sup> · Loïc Denis<sup>2</sup> · Florence Tupin<sup>3</sup>

Received: 16 June 2021 / Accepted: 31 December 2021 / Published online: 28 January 2022  
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Synthetic aperture radar (SAR) images are widely used for Earth observation to complement optical imaging. By combining information on the polarization and the phase shift of the radar echos, SAR images offer high sensitivity to the geometry and materials that compose a scene. This information richness comes with a drawback inherent to all coherent imaging modalities: a strong signal-dependent noise called “speckle.” This paper addresses the mathematical issues of performing speckle reduction in a transformed domain: the matrix-log domain. Rather than directly estimating noiseless covariance matrices, recasting the denoising problem in terms of the matrix-log of the covariance matrices stabilizes noise fluctuations and makes it possible to apply off-the-shelf denoising algorithms. We refine the method MuLoG by replacing heuristic procedures with exact expressions and improving the estimation strategy. This corrects a bias of the original method and should facilitate and encourage the adaptation of general-purpose processing methods to SAR imaging.

**Keywords** Covariance matrix · Denoising · Synthetic aperture radar · Plug-in ADMM · Estimation · Regularization

## 1 Introduction

SAR imaging is a key technology in airborne and satellite remote sensing. This active imaging technique based on time-of-flight measurement and coherent processing (the so-called aperture synthesis) has a night and day capability and can produce images through clouds [36]. Beyond intensity images, many SAR systems offer polarimetric information, i.e., they measure how the polarization of the wave is affected by the back-scattering mechanisms occurring when the electromagnetic radar wave interacts with the illuminated

scene. Backscattered radar waves collected from slightly different points of view can be combined, in a process called interferometry, to perform 3-D reconstructions or to recover very small displacements. In contrast to conventional optical imaging, SAR imaging gives access to both the amplitude *and the phase* of the backscattered wave. Interferometry uses this phase information to relate a phase shift observed between two SAR images to a change in the optical path (i.e., the path length traveled by the wave in its round-trip from the SAR antenna to the scene and back). Figure 1a illustrates the geometry of SAR imaging. Depending on the nature of the information available at each pixel, several names are used to describe SAR images:

✉ Charles-Alban Deledalle  
charles-alban.deledalle@math.u-bordeaux.fr

Loïc Denis  
loic.denis@univ-st-etienne.fr

Florence Tupin  
florence.tupin@telecom-paris.fr

<sup>1</sup> IMB, CNRS, Univ Bordeaux, Bordeaux INP, Talence 33405, France

<sup>2</sup> Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School, Laboratoire Hubert Curien UMR 5516, Saint-Etienne 42023, France

<sup>3</sup> LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France

- **SAR** denotes single-channel images: each pixel  $k$  contains the complex amplitude  $v_k \in \mathbb{C}$  backscattered by the scene. When no interferometric processing involving other single-channel SAR images is to be performed, the phase of  $v_k$  can be discarded and only the amplitude  $|v_k|$  or the intensity  $|v_k|^2$  is considered;
- **PolSAR** denotes multi-channel polarimetric images: each pixel  $k$  contains a vector  $\mathbf{v}_k = (v_k^1, v_k^2, v_k^3) \in \mathbb{C}^3$  of 3 complex amplitudes backscattered by the scene under different polarizations (horizontally or vertically

linearly polarized components at emission or reception, see Fig. 1a);

- **InSAR** denotes multi-channel interferometric images: each pixel  $k$  contains a vector  $\mathbf{v}_k \in \mathbb{C}^2$  formed by the 2 complex amplitudes backscattered by the scene on the two antenna positions, multi-baseline and tomographic SAR are extensions of SAR interferometry to more than two images;
- **PolInSAR** denotes multi-channel polarimetric and interferometric images: each pixel  $k$  contains a vector of 6 complex amplitudes  $\mathbf{v}_k \in \mathbb{C}^6$  and corresponds to the combination of polarimetric and interferometric information.

Polarimetric images are generally displayed in false colors using Pauli polarimetric basis, by combining the complex amplitudes collected under the different polarimetric configurations (the red channel corresponds to  $\frac{1}{2}|v_k^1 - v_k^3|^2$ , the green channel to  $2|v_k^2|^2$  and the blue channel to  $\frac{1}{2}|v_k^1 + v_k^3|^2$ ), see [36]. Figure 2 shows airborne images of the same area obtained with each SAR modality (image credit: ONERA). Different structures are visible in the images: cultivated fields, roads (appearing as dark lines in SAR images due to the low reflectivity of such smooth surfaces), trees and several farm buildings (in the bottom left quarter of the image). The SAR illumination comes from the left hand side of the images; shadows are thus visible on the right of all elevated elements (in particular, trees). A striking peculiarity of SAR images is the strong noise observed in all SAR modalities. This noise is unavoidable because it originates from the coherent illumination that is essential to the synthetic aperture processing. The SAR antenna collects at each time sample several echoes that interfere with each other, see Fig. 1b, c. The resulting complex amplitude  $\mathbf{v}_k \in \mathbb{C}^D$  is distributed, under the well-established speckle model due to Joseph Goodman [23], as a circular complex Gaussian distribution:

$$p_{\mathcal{V}}(\mathbf{v}_k; \boldsymbol{\Sigma}_k) = \frac{1}{\pi^D |\boldsymbol{\Sigma}_k|} \exp\left(-\mathbf{v}_k^* \boldsymbol{\Sigma}_k^{-1} \mathbf{v}_k\right), \quad (1)$$

where  $*$  denotes the conjugate transpose, and the complex-valued covariance matrix  $\boldsymbol{\Sigma}_k$  carries all the information about the backscattering process: in SAR imaging ( $D = 1$ ),  $\boldsymbol{\Sigma}_k$  corresponds to the reflectivity at pixel  $k$ , in PolSAR  $\boldsymbol{\Sigma}_k \in \mathbb{C}^{3 \times 3}$  characterizes the reflectivity in each polarimetric channel (diagonal of  $\boldsymbol{\Sigma}_k$ ) and the scattering mechanism (matrix  $\boldsymbol{\Sigma}_k$  can be decomposed into a sum of matrices corresponding to elementary phenomena such as surface scattering, dihedral scattering and volume scattering), in InSAR  $\boldsymbol{\Sigma}_k \in \mathbb{C}^{2 \times 2}$  and diagonal elements correspond to reflectivities while the off-diagonal elements indicate the phase shift  $\arg(\Sigma_k^{1,2})$  from one antenna to the other (due to the difference in path length)

and the coherence  $|\Sigma_k^{1,2}| / \sqrt{\Sigma_k^{1,1} \Sigma_k^{2,2}}$  (i.e., the remaining correlation between the complex amplitudes  $v_k^1$  and  $v_k^2$ : this correlation drops when the two images are captured at two dates that are more separate or when the scene has evolved between the two acquisitions, e.g., due to vegetation growth). Since the covariance matrix  $\boldsymbol{\Sigma}_k$  contains all the useful information, it has to be estimated from the diffusion vector  $\mathbf{v}_k$  to characterize the radar properties of the scene at each pixel. This is classically done by computing the sample covariance  $\mathbf{C}_k$  inside a small window centered at pixel  $k$ :

$$\mathbf{C}_k = \frac{1}{L} \sum_{\ell \in \mathcal{N}_k} \mathbf{v}_\ell \mathbf{v}_\ell^*, \quad (2)$$

where  $\mathcal{N}_k$  is the set of pixel indices within the window centered at pixel  $k$  and  $L = \text{Card}(\mathcal{N}_k)$  is the number of pixels in the window. If the speckle is spatially independent and all pixels in the window follow a distribution with a common covariance matrix  $\boldsymbol{\Sigma}_k$ , the samples  $\mathbf{v}_\ell$  are independent and identically distributed. When  $L \geq D$ , the sample covariance matrix is then distributed according to a complex Wishart distribution:  $\mathbf{C}_k \sim \mathcal{W}(\boldsymbol{\Sigma}_k, L)$ , and its multi-variate probability density function is given by [22]

$$p_{\mathcal{C}}(\mathbf{C}_k; \boldsymbol{\Sigma}_k, L) = \frac{L^D |\mathbf{C}_k|^{L-D}}{\Gamma_D(L) |\boldsymbol{\Sigma}_k|^L} \exp(-L \text{tr}(\boldsymbol{\Sigma}_k^{-1} \mathbf{C}_k)), \quad (3)$$

where  $\Gamma_D(L) = \pi^{D(D-1)/2} \prod_{k=1}^D \Gamma(L - k + 1)$ . In the case of single-channel SAR images ( $D = 1$ ),  $\mathbf{C}_k$  corresponds to an intensity  $I_k$  and  $\boldsymbol{\Sigma}_k$  is the pixel reflectivity  $R_k$ . The SAR intensity is then distributed according to a gamma distribution:

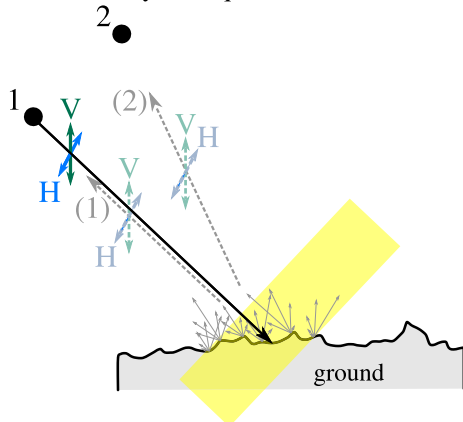
$$p_{\mathcal{I}}(I_k; R_k, L) = \frac{L^L I_k^{L-1}}{\Gamma(L) R_k^L} \exp(-L I_k / R_k). \quad (4)$$

The speckle in SAR intensity images is known to be a multiplicative noise in the sense that  $\text{Var}[I_k] = R_k^2 / L$ , so that the standard deviation of speckle fluctuations is proportional to the pixel reflectivity, and the speckle-corrupted intensity  $I_k$  may be modeled using a generative model of the form:  $I_k = R_k S_k$  with  $S_k$  a random variable distributed according to a gamma distribution with a mean equal to 1 and a variance equal to  $1/L$ . In the multi-variate case ( $D > 1$ ), the generative model becomes  $\mathbf{C}_k = \boldsymbol{\Sigma}_k^{1/2} \mathbf{S}_k \boldsymbol{\Sigma}_k^{1/2}$  with  $\mathbf{S}_k \sim \mathcal{W}(\mathbf{I}, L)$ , see [14]. The variance of  $C_k^{i,j}$  (the element of matrix  $\mathbf{C}_k$  at row  $i$  and column  $j$ ) is equal to  $\frac{1}{L} \Sigma_k^{i,i} \Sigma_k^{j,j}$ , i.e., the standard deviation of  $C_k^{i,j}$  is proportional to the geometric mean of the reflectivities in channels  $i$  and  $j$  and is thus also signal-dependent.

The ubiquity of speckle noise in SAR images, the multiplicative nature of the fluctuations and heavy-tailed behavior

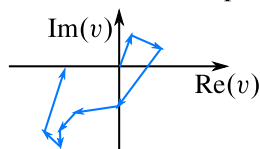
### The physics of SAR imaging:

#### (a) Geometry of acquisition

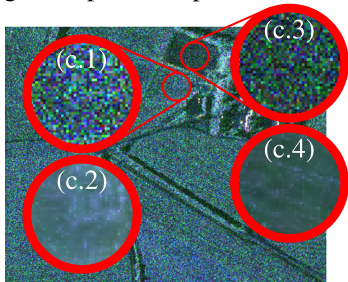


### The statistics of speckle:

#### (b) Coherent summation of radar echoes as a random walk in the complex plane



#### (c) signal-dependent speckle fluctuations:

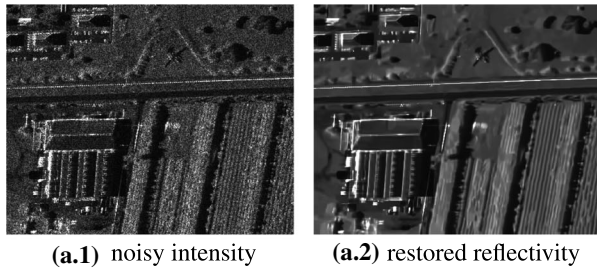
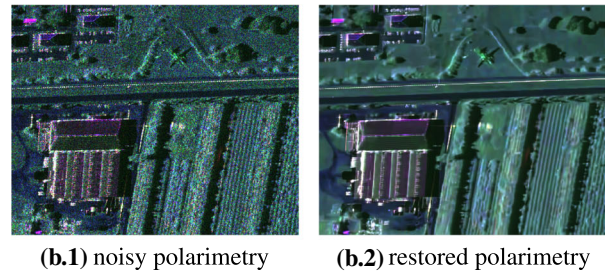
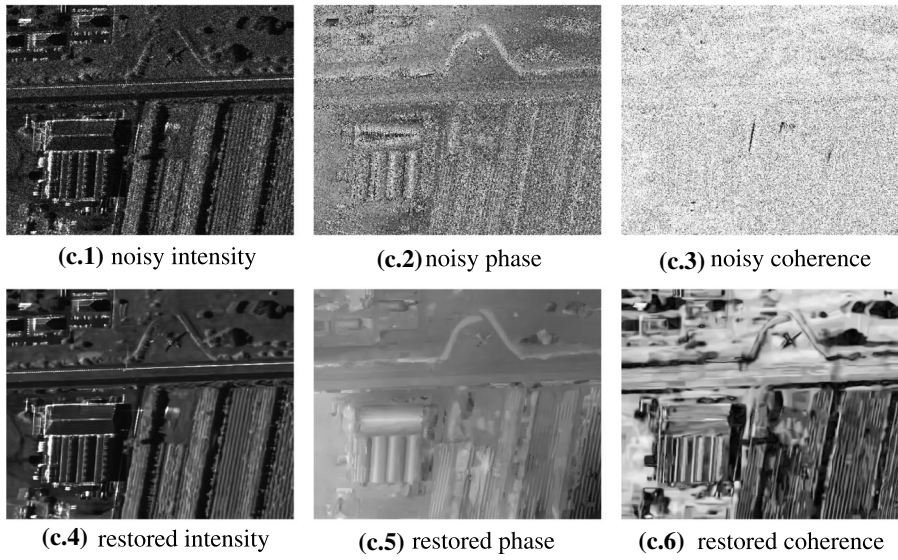
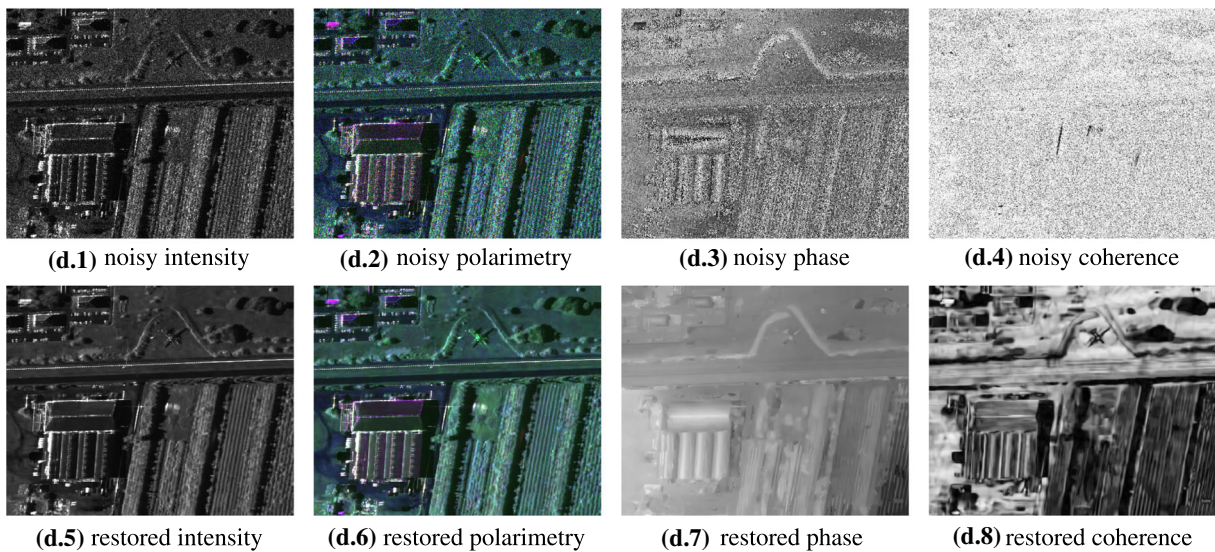


SAR imaging systems use antennas to emit radio waves (with wavelengths ranging from a few centimeters to a meter) and then collect the wave back-scattered by the illuminated scatterers, see (a). When polarimetric information is collected, the radar antenna alternatively emits vertically then horizontally linearly polarized waves. The vertical and horizontal components of the backscattered wave are then measured. In SAR interferometry, the back-scattered wave is collected on two antennas (represented as black dots 1 and 2), either simultaneously (mono-pass interferometry, depicted in the figure) or with a single-antenna system at two different times (multiple-pass interferometry). All the scatterers located within the rectangular region highlighted in the Figure produce radar echoes that are collected simultaneously on the antenna. These echoes interfere and the resulting complex amplitude measured by the antenna is highly fluctuating, depending on the destructive and constructive interferences. The simplest SAR systems do not record polarimetric or interferometric information, only a single complex amplitude  $v_k$  is measured at pixel  $k$ . Polarimetric systems collect the back-scattered amplitude in different configurations: for a horizontally linearly polarized emission and horizontally linearly polarized reception (HH), the cross-polarization corresponding to a horizontal emission and vertical reception (HV), or the converse (VH), and for vertically polarized waves in emission and reception (VV). Due to the symmetry of the system, HV and VH measurements are very similar and only one of them is generally considered. A polarimetric SAR image thus contains a vector  $\mathbf{v}_k \in \mathbb{C}^3$  at pixel  $k$ . SAR interferometry combines echoes collected for two distinct locations of the antenna. Two complex amplitudes are collected per pixel:  $\mathbf{v}_k \in \mathbb{C}^2$ . Polarimetry and interferometry can be combined, the diffusion vector  $\mathbf{v}_k$  then contains 6 complex amplitudes:  $\mathbf{v}_k \in \mathbb{C}^6$ .

The term “speckle” denotes the strong fluctuations of SAR amplitudes from one pixel to the next in homogeneous areas. Due to the coherent illumination, the complex amplitudes of the echoes returned by all the scatterers seen within a given pixel interfere. This corresponds to a random walk in the complex plane, depicted in (b). By the central limit theorem, it follows that, in the limit of a large number of independent elementary scatterers each introducing a uniformly distributed phase shift in  $]-\pi, \pi]$ , the resulting complex amplitude is distributed according to a circular complex Gaussian [23]. Speckle is a signal-dependent phenomenon: it is stronger in areas with larger reflectivities. This is illustrated in (c) where a polarimetric SAR image is displayed. Regions (c.1) and (c.3) correspond to two homogeneous areas with different reflectivities. Stronger fluctuations can be noted in area (c.1), whose underlying reflectivity is larger (the estimated reflectivity is shown in (c.2)), while region (c.3) displays smaller fluctuations, in line with a lower reflectivity (the estimated reflectivity of this region is shown in (c.4)).

**Fig. 1** A short introduction to the physics of SAR imaging and to the speckle phenomenon



$D = 1$ : SAR intensity images $D = 3$ : PolSAR polarimetric images $D = 2$ : InSAR interferometric images $D = 6$ : PolInSAR polarimetric and interferometric images

**Fig. 2** Synthetic aperture radar imaging offers rich information of a scene but suffers from speckle. The combination of images acquired with slightly different incidence angles (InSAR) or various polarization states (PolSAR) leads to a  $D$ -dimensional complex-valued vector per pixel. Reduction of the speckle fluctuations requires appropriate statistical modeling. This paper is devoted to the mathematical analysis of a generic approach for speckle reduction based on matrix-log decom-

positions. The images shown were obtained with the X-band airborne imaging system SETHI of the French aerospace lab ONERA [2] (after our pre-processing to achieve a trade-off between sidelobe attenuation and speckle decorrelation, the pixel size is  $\approx 70\text{cm} \times 70\text{cm}$ , and the area shown is  $\approx 300\text{m} \times 370\text{m}$ ). The Gaussian denoiser used in the despeckling algorithm is BM3D [10]

of Wishart distribution have fueled numerous developments of specific image processing methods to reduce speckle. The vast majority of these works considered single-channel SAR images, and adapted techniques based on pixel selection (based on range selection with Lee's  $\sigma$ -filter [30], window selection [32] or region growing [47]), variational techniques (total variation minimization [1, 19, 44], curvelets [20]), patch-based methods derived from the non-local means [7, 15–17, 38], or more recently deep-learning based methods (using supervised [8], semi-supervised [12] or self-supervised [34] learning strategies). The adaptation of a novel image denoising technique to the specificities of SAR imaging is a thorough process that includes replacing steps to account for the statistics of speckle and the nature of SAR images where many bright structures reach intensities several orders of magnitude larger than the surrounding area. This slows down the transfer of successful denoising techniques to the field of SAR imaging. In order to circumvent this adaptation process, we recently proposed a generic framework [14], named MuLoG, derived from the general “plug-and-play ADMM” strategy [6, 48] which is related to a wider family of approaches using denoisers to regularize inverse problems [40]. When an image restoration problem is stated in the form of a variational problem and then solved using the alternating directions method of multipliers (ADMM, see for example [3]), or proximal-splitting techniques [9], one step of the algorithm that improves the fidelity to the prior model corresponds to the denoising of an image corrupted by additive white Gaussian noise. The key idea of “plug-and-play ADMM” is then to replace this step by an off-the-shelf Gaussian denoiser.

The flexibility of MuLoG with respect to various SAR modalities (see the despeckling results obtained with MuLoG in Fig. 2) and its ability to benefit from the latest developments in additive Gaussian denoising makes the method very useful for SAR applications (e.g., multi-temporal filtering [53], deformation analysis [21], height retrieval [52] or despeckling using pre-trained neural networks [13]). The original MuLoG algorithm in [14] is based on approximations that can lead, however, to estimation biases. This paper starts with a brief summary of MuLoG framework in Sect. 2. We then perform a rigorous analysis of the optimization problem involved and establish the exact closed-form expression for the first and second directional derivatives of the matrix exponential mapping. We discuss the important problem of initialization and regularization of the covariance matrices, in particular in the rank-deficient case  $L < D$ . We introduce several modifications and show that they suppress the bias of the original method. Beyond their use in MuLoG's generic framework, these mathematical developments can benefit other variational methods for the restoration or segmentation of multi-channel SAR images, as well as hybrid methods that combine deep learning and an explicit statistical model of

speckle by algorithm unrolling [35]. For easier reproducibility, the source code of the algorithms described in this paper is made available at <https://www.charles-deledalle.fr/pages/mulog.php> under an open source license.

## 2 An Overview of MuLoG Framework

In order to give a self-contained presentation of our developments, we recall in this section the principle of MuLoG, as first introduced in [14]. MuLoG's approach to multi-channel SAR despeckling is built around two key ideas:

- a nonlinear transform that decomposes the field of  $D \times D$  noisy covariance matrices  $\{\mathbf{C}_k\}_{k=1..n}$  into  $D^2$  real-valued images with  $n$ -pixels; this transform approximately stabilizes speckle fluctuations and decorrelates the channels so that each can be denoised independently;
- implicit regularization using a plug-and-play ADMM iterative scheme where the proximal operator associated to the prior term is replaced by an off-the-shelf Gaussian denoiser.

The nonlinear transform is defined in three steps: (i) a matrix-log is applied to map each speckled covariance matrix  $\mathbf{C}_k$  to a Hermitian matrix with approximately stabilized variance; (ii) the real and imaginary parts of these Hermitian matrices are separated, forming  $D^2$  real-valued channels; (iii) an affine transform, identical for all pixel locations  $k$ , whitens these channels. The noisy covariance matrix  $\mathbf{C}_k \in \mathbb{C}^{D \times D}$  is then re-parameterized by  $\mathbf{y}_k \in \mathbb{R}^{D^2}$ , and similarly the covariance matrix of interest  $\boldsymbol{\Sigma}_k$  is defined through the real-valued vector  $\mathbf{x}_k \in \mathbb{R}^{D^2}$ :

$$\mathbf{C}_k = e^{\Omega(\mathbf{y}_k)} \quad \text{and} \quad \boldsymbol{\Sigma}_k = e^{\Omega(\mathbf{x}_k)} \quad (5)$$

where the exponential corresponds to a matrix-exponential and the affine invertible mapping  $\Omega: \mathbb{R}^{D^2} \rightarrow \mathbb{C}^{D \times D}$  can be decomposed as  $\Omega(\mathbf{x}) = \mathcal{K}(\mathbf{W}\boldsymbol{\Phi}\mathbf{x} + \mathbf{b})$ , with  $\mathcal{K}: \mathbb{R}^{D^2} \rightarrow \mathbb{C}^{D \times D}$  the linear operator that transforms a vector of  $D^2$  reals into a  $D \times D$  Hermitian matrix:

$$\mathcal{K} \begin{pmatrix} \alpha^1 \\ \vdots \\ \alpha^{D^2} \end{pmatrix} = \begin{pmatrix} \alpha^1 & \frac{\alpha^{D+1} + j\alpha^{D+2}}{\sqrt{2}} & \dots & \frac{\alpha^{D^2-1} + j\alpha^{D^2}}{\sqrt{2}} \\ \frac{\alpha^{D+1} - j\alpha^{D+2}}{\sqrt{2}} & \alpha^2 & & \\ \vdots & & \ddots & \vdots \\ \frac{\alpha^{D^2-1} - j\alpha^{D^2}}{\sqrt{2}} & & \dots & \alpha^D \end{pmatrix}, \quad (6)$$



$\mathbf{W} \in \mathbb{R}^{D^2 \times D^2}$  a (whitening) unitary matrix,  $\Phi \in \mathbb{R}^{D^2 \times D^2}$  a diagonal positive definite matrix (used to weight each channel) and  $\mathbf{b} \in \mathbb{R}^{D^2}$  a real vector (for centering). To compute the real-valued decomposition  $\mathbf{x}$  of a covariance matrix  $\Sigma$ , the inverse transform can be applied:  $\mathbf{x} = \Phi^{-1} \mathbf{W}^{-1} (\mathcal{K}^{-1}(\log(\Sigma)) - \mathbf{b})$ . A principal component analysis is used to compute matrix  $\mathbf{W}$  and vector  $\mathbf{b}$ :

$$b_i = \frac{1}{n} \sum_{k=1}^n \alpha_k^i, \tag{7}$$

with  $\alpha_k^i = \mathcal{K}^{-1}(\log(C_k))$ , the  $i$ -th real-value extracted from the log-transformed covariance  $\tilde{C}_k = \log(C_k)$ , and the columns of matrix  $\mathbf{W}$  are unit-norm eigenvectors of the Gram matrix:

$$\frac{1}{n} \sum_{k=1}^n (\alpha_k^i - b_i)(\alpha_k^j - b_j). \tag{8}$$

The  $i$ -th diagonal element of  $\Phi$  corresponds to the noise standard deviation of the  $i$ -th channel estimated via the median absolute deviation (MAD) estimator.

Figure 3 illustrates the channels obtained with the transform  $\Omega^{-1}$ . Fluctuations due to the speckle noise have a variance that is approximately stabilized in the channels of  $\mathbf{y}$ . Due to the whitening with matrix  $\mathbf{W}$ , most of the information is captured by the first channels (associated with the largest eigenvalues) and the signal-to-noise ratio decreases with the channel number. In the last channels, some denoising artifacts can be noticed (e.g., bottom right image of Fig.3). These artifacts have a negligible impact because of the small contribution of the channel to the recomposed image.

The neg-log-likelihood of the re-parameterized covariance matrix can be derived from the Wishart distribution given in Eq. (3):

$$-\log p_{\mathbf{y}}(\mathbf{y}_k | \mathbf{x}_k) = L \operatorname{tr}[\Omega(\mathbf{x}_k) + e^{-\Omega(\mathbf{x}_k)} e^{\Omega(\mathbf{y}_k)}] + \text{Cst}, \tag{9}$$

where  $\mathbf{x}_k$  and  $\mathbf{y}_k$  are the  $D^2$ -dimensional real-valued vectors corresponding, respectively, to the re-parameterization of the noiseless and noisy covariance matrices, while Cst is a constant independent of  $\mathbf{x}_k$ .

MuLoG is detailed in Algorithm 1. In practice, a fixed number of steps are generally used to stop the main loop (typically 6 steps provide a good trade-off between computational cost and restoration quality). Parameter  $\beta$  is increased within the loop, line 11, to ensure the convergence (we use the adaptive update rule given in Eq. (15) of paper [6] which consists of increasing  $\beta$  whenever the norm of the primal and dual residuals decreases too slowly between two successive

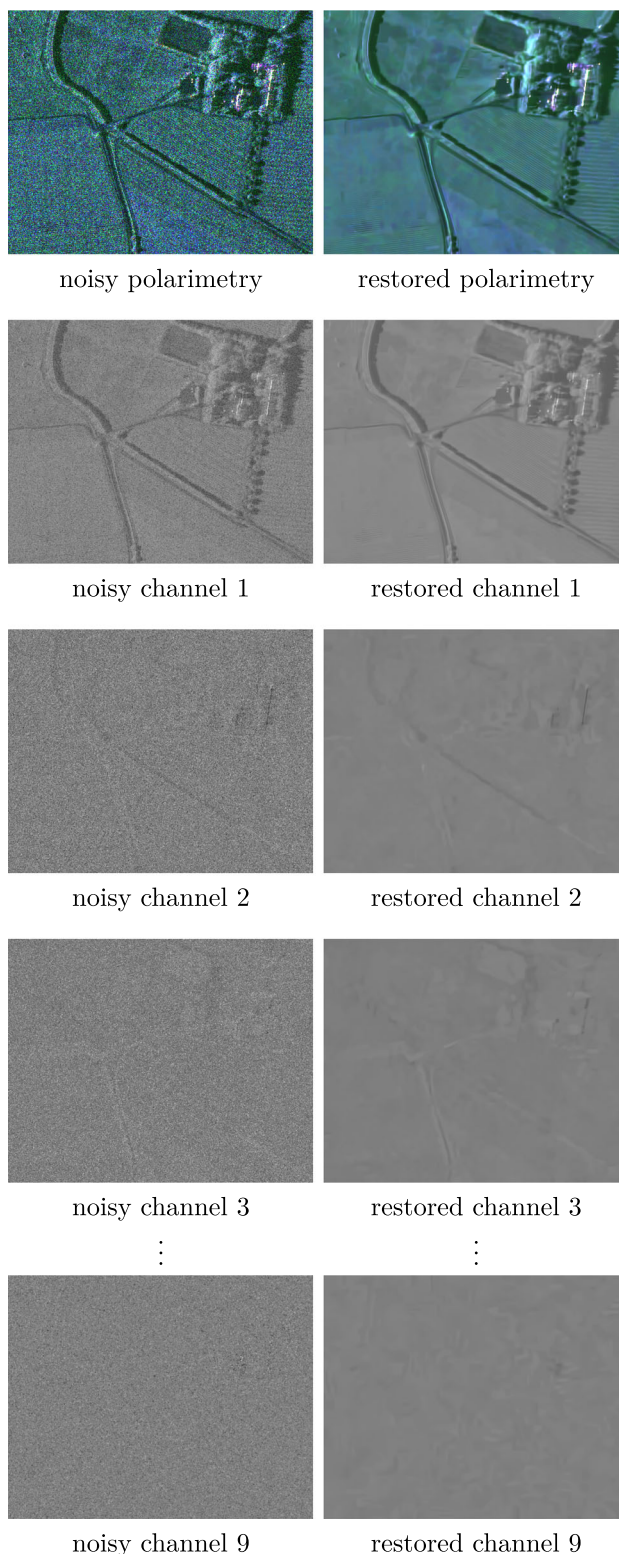


Fig. 3 Illustration of the matrix-log decomposition of a polarimetric SAR image into 9 real-valued channels (airborne image from ONERA-SETHI, see the caption of Fig. 2)

steps). Convergence of plug-in ADMM schemes has been studied in several works, e.g.,

[6,25,43,45], under various assumptions on the denoiser (non-expansiveness, boundedness), the data fidelity (convexity) or the evolution of parameter  $\beta$ . In our case, the data fidelity is non-convex and the convergence comes from the continuity of the gradient, which implies that the gradient is Lipschitz on any compact subset and that the conditions in [6] are fulfilled provided that the denoiser is bounded. Adaptations of the loss function or normalization procedures have been proposed for denoisers based on deep neural networks in [43,45] to ensure that the denoiser is firmly non-expansive, which also ensures convergence [6]. Several steps playing an important role in MuLoG algorithm are revisited in this paper:

- the minimization technique to compute, at each pixel  $k$ , the  $D^2$ -dimensional vector  $\hat{\mathbf{x}}_k$  line 10: this paper develops in Sect. 3 a new algorithm based on exact derivatives to improve this crucial step;
- the initial estimate used for  $\hat{\mathbf{x}}_k$ , line 4, computed from the matrices  $\{C_k^{(\text{init})}\}_{k=1\dots,n}$  defined line 1: an adequate initialization speeds up the convergence as discussed in Sect. 4.1;
- the regularization of the noisy covariance matrices, set line 2, that impacts the minimization problem line 10: regularization strategies can avoid badly behaved cost functions that are hard to minimize but can also lead to estimation biases, see Sect. 4.2;
- the choice of the denoiser used line 8: as illustrated in the discussion (Sect. 6), each denoiser suffers from specific artifacts.

### 3 Improved Computation of the Data-Fidelity Proximal Operator

Line 10 of Algorithm 1 involves solving  $n$  independent  $D^2$ -dimensional minimization problems of an objective function of the form

$$F(\mathbf{x}) = \frac{\beta}{2} \|\mathbf{x} - \mathbf{u}\|^2 + L \text{tr}(\Omega(\mathbf{x}) + e^{\Omega(\mathbf{y})} e^{-\Omega(\mathbf{x})}). \quad (10)$$

This corresponds to computing the data-fidelity proximal operator [9]:

$$\text{prox}_{\text{data}}(\mathbf{u}) = \arg \min_{\mathbf{x}} \underbrace{\frac{\beta}{2} \|\mathbf{x} - \mathbf{u}\|^2 - \log p_{\mathcal{Y}}(\mathbf{y}|\mathbf{x})}_{F(\mathbf{x})}, \quad (11)$$

where  $\mathbf{x}$ ,  $\mathbf{y}$  and  $\mathbf{u}$  are vectors in  $\mathbb{R}^{D^2}$ . We recall that  $\mathbf{y}$  corresponds to the noisy log-channels defined in Eq. (5) and

---

#### Algorithm 1: MuLoG algorithm

---

**input** : a bidimensional field of speckle-corrupted covariance matrices  $\mathbf{C}$

**input** : a denoiser function  $f_{\sigma} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  to remove additive white Gaussian noise with standard deviation  $\sigma$  in  $n$ -pixels single-channel images

**output**: a bidimensional field of estimated covariance matrices  $\hat{\Sigma}$

**initialization**:

- 1 estimate  $\{C_k^{(\text{init})}\}_{k=1..n}$  (initial estimation)  
 $\hookrightarrow$  **Algorithm 4**
- 2 estimate  $\{C_k^{(\text{reg})}\}_{k=1..n}$  (enforces PSD matrices)  
 $\hookrightarrow$  **Algorithm 5**
- 3 compute  $\mathbf{b}$  and  $\mathbf{W}$  (Eqs.(7) and (8))
- 4  $\{\hat{\mathbf{x}}_k \leftarrow \Omega^{-1}(\log C_k^{(\text{init})})\}_{k=1..n}$  (initial estimation)
- 5  $\{\mathbf{y}_k \leftarrow \Omega^{-1}(\log C_k^{(\text{reg})})\}_{k=1..n}$  (noisy channels)
- 6  $\{\mathbf{d}_k \leftarrow \mathbf{0}_{D^2}\}_{k=1..n}$  (initial scaled multipliers)
- 7  $\beta \leftarrow 1 + \frac{2}{L}$  (initial value)

**repeat**

- 8  $\{\hat{\mathbf{z}}^i \leftarrow f_{\beta^{-1/2}}(\hat{\mathbf{x}}^i - \hat{\mathbf{d}}^i)\}_{i=1..D^2}$  (denoise channels)
- 9  $\hat{\mathbf{d}} \leftarrow \hat{\mathbf{d}} + \hat{\mathbf{z}} - \hat{\mathbf{x}}$  (update multipliers)
- 10  $\{\hat{\mathbf{x}}_k \leftarrow \arg \min_{\mathbf{x}_k} \frac{\beta}{2} \|\mathbf{x}_k - \hat{\mathbf{z}}_k - \hat{\mathbf{d}}_k\|^2 - \log p_{\mathcal{Y}}(\mathbf{y}_k|\mathbf{x}_k)\}_{k=1..n}$  (data-fidelity prox)  
 $\hookrightarrow$  **Algorithm 3**
- 11 update  $\beta$

**until convergence**;

- 12  $\{\hat{\Sigma}_k \leftarrow e^{\Omega(\hat{\mathbf{x}}_k)}\}_{k=1..n}$  (final estimation)

---

shown in the first column of Fig. 3, while  $\mathbf{u} = \hat{\mathbf{z}}_k + \hat{\mathbf{d}}_k$  in line 10 of Algorithm 1. Efficient minimization methods require the computation of the gradient  $\mathbf{g} = \nabla F(\mathbf{x})$  of  $F$  at  $\mathbf{x}$ .

As shown in [14] and recalled in Appendix A for the sake of completeness, this gradient is given by

$$\mathbf{g} = \beta(\mathbf{x} - \mathbf{z}) + L\Theta \left( \text{Id}_D - \frac{\partial e^{\Sigma}}{\partial \Sigma} \Big|_{-\Omega(\mathbf{x})}^* [e^{\Omega(\mathbf{y})}] \right), \quad (12)$$

where the linear operator  $\Theta$  is defined by

$$\Theta(\cdot) = \Phi \mathbf{W}^* \mathcal{K}^*(\cdot) \quad (13)$$

and  $\frac{\partial e^{\Sigma}}{\partial \Sigma} \Big|_{-\Omega(\mathbf{x})}^* [e^{\Omega(\mathbf{y})}]$  is the adjoint of the directional derivative of the matrix exponential in the direction  $e^{\Omega(\mathbf{y})}$  taken at

$-\Omega(\mathbf{x})$ . We recall that the directional derivative of a differentiable function  $f$  at  $\mathbf{X}$  in the direction  $\mathbf{A}$  is defined as

$$\left. \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \right|_{\mathbf{X}} [\mathbf{A}] = \lim_{\epsilon \rightarrow 0} \frac{1}{\epsilon} (f(\mathbf{X} + \epsilon \mathbf{A}) - f(\mathbf{X})) . \tag{14}$$

The directional derivative is a linear mapping with respect to the direction  $\mathbf{A}$ . The computation of the gradient thus requires computing the adjoint of such linear mapping for the matrix exponential.

In the original derivation of MuLoG algorithm [14], we used an integral expression for the adjoint of the directional derivative of the matrix exponential

$$\left. \frac{\partial e^{\Sigma}}{\partial \Sigma} \right|_{\Sigma}^* [\mathbf{A}] = \int_0^1 e^{u\Sigma} \mathbf{A} e^{(1-u)\Sigma} du \tag{15}$$

to derive an approximation based on a Riemann sum:

$$\left. \frac{\partial e^{\Sigma}}{\partial \Sigma} \right|_{\Sigma}^* [\mathbf{A}] \approx \frac{1}{Q} \sum_{q=1}^Q e^{u_q \Sigma} \mathbf{A} e^{(1-u_q)\Sigma} \tag{16}$$

where  $u_q = (q - 1/2)/Q$ . In practice, we were using only  $Q = 1$  rectangle to get a fast-enough approximation. In the next paragraph, we derive a closed-form expression of the gradient to avoid this approximation, and then, we obtain the closed-form expression of second-order directional derivative of  $F$  in order to obtain an improved minimization method for the computation of  $\text{prox}_{\text{data}}$  in Eq. (11).

### 3.1 Closed-Form Expression of the Gradient

We leverage studies from [5,18,31] regarding the derivatives of matrix spectral functions. As shown in ‘‘Appendix B,’’ these leads to a closed-form expression for this directional derivative of the matrix exponential as described in the next proposition.

**Proposition 1** *Let  $\Sigma$  be a  $D \times D$  Hermitian matrix with distinct eigenvalues. Let  $\Sigma = \mathbf{E} \text{diag}(\Lambda) \mathbf{E}^*$  be its eigen-decomposition where  $\mathbf{E}$  is a unitary matrix of eigenvectors and  $\Lambda = (\lambda_1, \dots, \lambda_D)$  the vector of corresponding eigenvalues. Then, for any  $D \times D$  Hermitian matrix  $\mathbf{A}$ , denoting  $\bar{\mathbf{A}} = \mathbf{E}^* \mathbf{A} \mathbf{E}$ , we have*

$$\left. \frac{\partial e^{\Sigma}}{\partial \Sigma} \right|_{\Sigma} [\mathbf{A}] = \mathbf{E} [\mathbf{G} \odot \bar{\mathbf{A}}] \mathbf{E}^* \tag{17}$$

where  $\odot$  is the element-wise (a.k.a, Hadamar) product, and, for all  $1 \leq i, j \leq D$ , we have defined

$$\mathbf{G}_{i,j} = \begin{cases} \frac{e^{\lambda_i} - e^{\lambda_j}}{\lambda_i - \lambda_j} & \text{if } i \neq j, \\ e^{\lambda_i} & \text{otherwise.} \end{cases} \tag{18}$$

---

#### Algorithm 2: Exact evaluation of the gradient $\nabla F$

---

**input** : a vector  $\mathbf{x} \in \mathbb{R}^{D^2}$   
**input** : a vector  $\mathbf{y} \in \mathbb{R}^{D^2}$   
**input** : a vector  $\mathbf{u} \in \mathbb{R}^{D^2}$   
**input** : the affine map  $\Omega : \mathbf{x} \mapsto \mathcal{K}(\mathbf{W}\Phi\mathbf{x} + \mathbf{b})$   
**output**: the gradient  $\mathbf{g}$  of  $F(\mathbf{x})$  defined in (10)

- 1 compute the eigenvalue decomposition:  
 $-\Omega(\mathbf{x}) = \mathbf{E} \text{diag}(\Lambda) \mathbf{E}^*$
- 2 compute  $\mathbf{G}$  (Eq.(18))
- 3  $\{\mathbf{G}_{i,j} \leftarrow \text{Proj}(\mathbf{G}_{i,j})\}_{i,j=1..D^2}$  (projection on  $[e^{\lambda_j}, e^{\lambda_i}]$ )
- 4  $\mathbf{A} \leftarrow e^{\Omega(\mathbf{y})}$
- 5  $\bar{\mathbf{A}} \leftarrow \mathbf{E}^* \mathbf{A} \mathbf{E}$
- 6  $\mathbf{g} \leftarrow \beta(\mathbf{x} - \mathbf{u}) + L\Theta(\text{Id}_D - \mathbf{E}[\mathbf{G} \odot \bar{\mathbf{A}}] \mathbf{E}^*)$

---

Note that Proposition 1 assumes the eigenvalues of  $\Sigma$  to be distinct. In practice, we observe instabilities when  $\lambda_i$  and  $\lambda_j$  are close to each other. To solve this numerical issue, we use that, for  $\lambda_i > \lambda_j$ ,

$$e^{\lambda_j} \leq \frac{e^{\lambda_i} - e^{\lambda_j}}{\lambda_i - \lambda_j} \leq e^{\lambda_i} \tag{19}$$

since exp is convex and increasing. Then, whenever an off-diagonal element  $\mathbf{G}_{i,j}$  is out of this constraint we project its value onto the feasible range<sup>1</sup>  $[e^{\lambda_j}, e^{\lambda_i}]$ . Equation (19) shows that  $\lim_{\lambda_j \rightarrow \lambda_i} \frac{e^{\lambda_i} - e^{\lambda_j}}{\lambda_i - \lambda_j} = e^{\lambda_i}$ . In the case of duplicate eigenvalues, we use the continuous expansion obtained by replacing the condition  $i \neq j$  with  $\lambda_i \neq \lambda_j$ . We checked numerically that this continuous expansion is working for matrices with repeated eigenvalues.

Proposition 1 gives us an exact closed-form formula for the directional derivative. The next corollary shows that this formula is also valid for its adjoint (see proof in ‘‘Appendix C’’).

**Corollary 1** *Let  $\Sigma$  be a Hermitian matrix with distinct eigenvalues. The Jacobian of the matrix exponential is a self-adjoint operator*

$$\left. \frac{\partial e^{\Sigma}}{\partial \Sigma} \right|_{\Sigma}^* = \left. \frac{\partial e^{\Sigma}}{\partial \Sigma} \right|_{\Sigma} . \tag{20}$$

Based on the closed-form expressions provided by Proposition 1 and Corollary 1, we define Algorithm 2 for an exact evaluation of the gradient of  $F$ .

---

<sup>1</sup> The projection is applied as follows:  $\mathbf{G}_{i,j} \leftarrow e^{\lambda_i}$  if  $\mathbf{G}_{i,j} \notin [e^{\lambda_j}, e^{\lambda_i}]$  and  $|\mathbf{G}_{i,j} - e^{\lambda_i}| < |\mathbf{G}_{i,j} - e^{\lambda_j}|$ , or  $\mathbf{G}_{i,j} \leftarrow e^{\lambda_j}$  if  $\mathbf{G}_{i,j} \notin [e^{\lambda_j}, e^{\lambda_i}]$  and  $|\mathbf{G}_{i,j} - e^{\lambda_i}| > |\mathbf{G}_{i,j} - e^{\lambda_j}|$



### 3.2 A Refined Quasi-Newton Scheme

The computation of the proximal operator requires the minimization of function  $F(\mathbf{x})$ . This can be performed using quasi-Newton iterations:

$$\mathbf{x} \leftarrow \mathbf{x} - \hat{\mathbf{H}}^{-1} \mathbf{g} \tag{21}$$

where  $\hat{\mathbf{H}}$  is an approximation of the Hessian  $\mathbf{H}$  of  $F$  at  $\mathbf{x}$ , i.e., the real symmetric matrix defined by

$$\mathbf{H} = \frac{\partial^2 F(\mathbf{x})}{\partial \mathbf{x}^2} = \frac{\partial}{\partial \mathbf{x}} \nabla_{\mathbf{x}} F(\mathbf{x}). \tag{22}$$

While in [14] a heuristic was used to define a diagonal approximation to the matrix  $\mathbf{H}$ , we consider here the following approximation:

$$\hat{\mathbf{H}} = \gamma \text{Id}_{D^2} \text{ with } \gamma = \mathbf{d}^* \mathbf{H} \mathbf{d} \text{ and } \mathbf{d} = \frac{\mathbf{g}}{\|\mathbf{g}\|} \tag{23}$$

where  $\gamma$  corresponds to the *exact* second derivative of  $F$  at  $\mathbf{x}$  in the direction  $\mathbf{d}$  of the gradient  $\mathbf{g}$ .

As proved in Appendix D, this second-order derivative is given by:

$$\gamma = \beta + L \left\langle \Theta^*(\mathbf{d}), \left. \frac{\partial^2 e^{\Sigma}}{\partial \Sigma^2} \right|_{-\Omega(\mathbf{x})} [e^{\Omega(\mathbf{y})}, \Theta^*(\mathbf{d})] \right\rangle, \tag{24}$$

where  $\Theta^*(\cdot) = \mathcal{K}(\mathbf{W}\Phi \cdot)$  is the adjoint of linear operator  $\Theta$  and for any matrices  $\mathbf{X}$  and  $\mathbf{Y}$ , the matrix dot product is defined as  $\langle \mathbf{X}, \mathbf{Y} \rangle = \text{tr}(\mathbf{X}^* \mathbf{Y})$ . This choice for approximating the Hessian leads to a quasi-Newton step that is exact when restricted to the direction of the gradient. As  $F$  has some regions of non-convexity, we consider in practice half the absolute value of the scalar product in (24) in order to avoid some local minima and ensure that quasi-Newton follows a descent direction. Note that, like with the original formulation of MuLoG, we recover the exact Newton method in the mono-channel case ( $D = 1$ ).

The computation of  $\gamma$  thus requires computing the second-order directional derivative of the matrix exponential. We recall that the second-order directional derivative of a function  $f : \mathbb{C}^{D \times D} \rightarrow \mathbb{C}^{D \times D}$  at  $\mathbf{X}$  in the directions  $\mathbf{A}$  and  $\mathbf{B}$  is the  $D \times D$  complex matrix defined as

$$\begin{aligned} & \frac{\partial^2 f(\mathbf{X})}{\partial \mathbf{X}^2} [\mathbf{A}, \mathbf{B}] \tag{25} \\ &= \lim_{\epsilon \rightarrow 0} \frac{1}{4\epsilon^2} [f(\mathbf{X} + \epsilon \mathbf{A} + \epsilon \mathbf{B}) - f(\mathbf{X} - \epsilon \mathbf{A} + \epsilon \mathbf{B}) \\ & \quad - f(\mathbf{X} + \epsilon \mathbf{A} - \epsilon \mathbf{B}) + f(\mathbf{X} - \epsilon \mathbf{A} - \epsilon \mathbf{B})]. \end{aligned}$$

---

#### Algorithm 3: Iterative method to compute $\text{prox}_{\text{data}}$

---

**input** : a noisy vector  $\mathbf{y} \in \mathbb{R}^{D^2}$   
**input** : a vector  $\mathbf{u} \in \mathbb{R}^{D^2}$   
**input** : the affine map  $\Omega : \mathbf{x} \mapsto \mathcal{K}(\mathbf{W}\Phi \mathbf{x} + \mathbf{b})$   
**input** : the linear map  $\Theta : \alpha \mapsto \Phi \mathbf{W}^* \mathcal{K}^* \alpha$   
**output**: the approximate minimizer of  $F(\mathbf{x})$

- 1  $\mathbf{x} \leftarrow \frac{\beta}{\beta+1} \mathbf{u} + \frac{1}{\beta+1} \mathbf{y}$  (initialization)
- repeat**
- 2  $\mathbf{g} \leftarrow \text{Algorithm 2}(\mathbf{x}, \mathbf{y}, \mathbf{u}, \Omega)$  (gradient)
- 3  $\mathbf{d} \leftarrow \mathbf{g} / \|\mathbf{g}\|$  (gradient direction)
- 4  $\bar{\mathbf{B}} \leftarrow \mathbf{E}^* \Theta^*(\mathbf{d}) \mathbf{E}$
- 5 compute  $F(\bar{\mathbf{A}}, \bar{\mathbf{B}})$  (Eq.(27))
- 6  $\gamma \leftarrow \beta + L | \langle \bar{\mathbf{B}}, F(\bar{\mathbf{A}}, \bar{\mathbf{B}}) \rangle |$  (inverse step size)
- 7  $\mathbf{x} \leftarrow \mathbf{x} - \frac{1}{2\gamma} \mathbf{g}$  (quasi-Newton step)

**until** convergence;

---

The closed-form expression for the second-order directional derivative of the matrix exponential is given in the following Proposition (see proof in “Appendix E”):

**Proposition 2** *Let  $\Sigma$  be a  $D \times D$  Hermitian matrix with distinct eigenvalues. Let  $\Sigma = \mathbf{E} \text{diag}(\Lambda) \mathbf{E}^*$  be its eigen-decomposition where  $\mathbf{E}$  is a unitary matrix of eigenvectors and  $\Lambda = (\lambda_1, \dots, \lambda_D)$  the vector of corresponding eigenvalues. For any  $D \times D$  Hermitian matrices  $\mathbf{A}$  and  $\mathbf{B}$ , denoting  $\bar{\mathbf{A}} = \mathbf{E}^* \mathbf{A} \mathbf{E}$  and  $\bar{\mathbf{B}} = \mathbf{E}^* \mathbf{B} \mathbf{E}$ , we have*

$$\frac{\partial^2 e^{\Sigma}}{\partial \Sigma^2} [\mathbf{A}, \mathbf{B}] = \mathbf{E} [F(\bar{\mathbf{A}}, \bar{\mathbf{B}})] \mathbf{E}^* \tag{26}$$

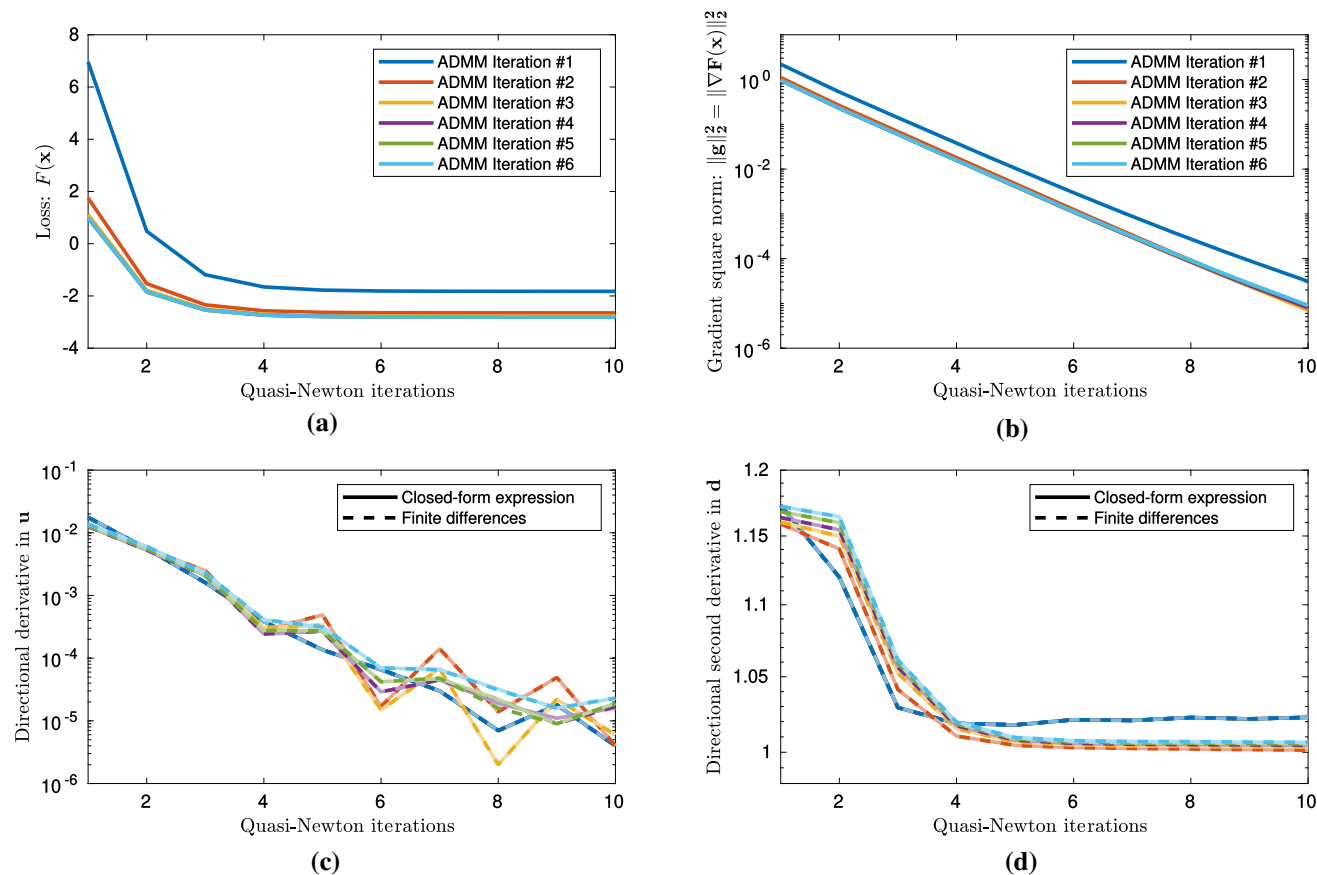
where, for all  $1 \leq i, j \leq D$ , we have

$$F(\bar{\mathbf{A}}, \bar{\mathbf{B}})_{i,j} = \sum_{k=1}^D \phi_{i,j,k} (\bar{\mathbf{A}}_{ik} \bar{\mathbf{B}}_{jk}^* + \bar{\mathbf{B}}_{ik} \bar{\mathbf{A}}_{jk}^*) \tag{27}$$

$$\text{with } \phi_{i,j,k} = \begin{cases} \frac{G_{ik} - G_{jk}}{\lambda_i - \lambda_j} & \text{if } i \neq j, \\ \frac{G_{ii} - G_{ik}}{\lambda_i - \lambda_k} & \text{if } i = j \text{ and } k \neq i, \\ \frac{G_{ii}}{2} & \text{if } i = j = k. \end{cases} \tag{28}$$

As in Proposition 1, Proposition 2 assumes the eigenvalues to be distinct. We checked on numerical simulations that the result is still valid for Hermitian matrices with duplicate eigenvalues by simply defining  $G_{ij}$  in Proposition 1 using the condition  $\lambda_i \neq \lambda_j$  instead of  $i \neq j$ .

Algorithm 3 details the quasi-Newton optimization scheme to solve the minimization problem (11). The algorithm starts with an initial value for  $\mathbf{x}$ , line 1, obtained by approximating the data likelihood with a Gaussian (i.e., Wiener).



**Fig. 4** Top row: evolution with respect to the quasi-Newton iterations of **a** the loss function  $F(\mathbf{x})$ , **b** the square norm of its gradient  $\mathbf{g} = \nabla F(\mathbf{x})$ . Bottom row: numerical validation of the closed-form expressions for the first-order **c** and second-order **d** directional derivatives  $\gamma$ : values

obtained with the closed-form expressions are drawn with solid lines; finite-differences derivatives are shown with dashed lines that superimpose closely. Curves are drawn with different colors for each of the first six iterations of the Plug-and-Play ADMM

### 3.3 Numerical Validation

In order to validate the correctness of the derived closed-form expressions of the gradient and the directional second derivative, we leverage that for  $\mathbf{g} = \nabla F(\mathbf{x})$  and  $\gamma = \mathbf{d}^* \frac{\partial^2 F(\mathbf{x})}{\partial \mathbf{x}^2} \mathbf{d}$ , the following two identities regarding the first and second directional derivatives hold true

$$\langle \mathbf{g}, \mathbf{u} \rangle = \lim_{\epsilon \rightarrow 0} \frac{F(\mathbf{x} + \epsilon \mathbf{u}) - F(\mathbf{x})}{\epsilon}, \quad \forall \mathbf{u} \in \mathbb{R}^{D^2}, \quad (29)$$

and

$$\gamma = \lim_{\epsilon \rightarrow 0} \frac{F(\mathbf{x} + \epsilon \mathbf{d}) - 2F(\mathbf{x}) + F(\mathbf{x} - \epsilon \mathbf{d})}{\epsilon^2}. \quad (30)$$

During the iterations of the proposed quasi-Newton scheme, we evaluate the left-hand sides of these equalities by running Algorithm 2 and 3 to compute  $\mathbf{g}$  and  $\gamma$ . Independently, we

evaluate the right-hand sides of these equalities by finite difference with a small nonzero value of  $\epsilon$ . The direction  $\mathbf{u}$  was chosen as a fixed white standardized normal vector.

Figure 4 shows the evolution of these four quantities during the iterations of the proposed quasi-Newton scheme for an arbitrary choice of the image  $\mathbf{y}$ , initializations and constant  $L$  and  $\beta$ . In addition, the evolution of  $F(\mathbf{x})$  and  $\|\mathbf{g}\|_2^2 = \|\nabla F(\mathbf{x})\|_2^2$  are also provided. On the one hand, we can notice that the proposed first and second directional derivatives are very close to the ones estimated by finite differences, which shows the validity of our formula. On the other hand, the objective  $F(\mathbf{x})$  is decreasing and its gradient converges to 0, showing that the obtained stationary point is likely to be a local minimum. Furthermore, one can notice that the second directional derivative varies less than 20% showing that the loss  $F$  is nearly quadratic in the vicinity of its minimizers.

### 4 Initialization and Regularization of Rank-Deficient Matrix Fields

Multi-channel SAR images with  $n$  pixels are provided in the form of a bidimensional field of either  $n$  diffusion vectors  $\mathbf{v}_k \in \mathbb{C}^D$  (single-look data) or  $n$  covariance matrices  $\mathbf{C}_k \in \mathbb{C}^{D \times D}$  (multi-look data), see Figure 1. The statistical distribution of speckle is defined with respect to full-rank covariance matrices  $\Sigma_k$ . The initial guess for this matrix must be positive-definite, which requires an adapted strategy. We discuss different strategies for this initialization in Paragraph 4.1. Then, we describe in Paragraph 4.2 how the noisy covariance matrices  $\mathbf{C}_k$  can be regularized so that the neg-log-likelihood is better behaved. We show that, in contrast to the original MuLoG method [14], different regularizations should be used for the initialization of the covariance matrices (line 1 of Algorithm 1) and for the computation of the noisy covariance matrices used to define the data-fidelity proximal operator (line 2 of Algorithm 1).

#### 4.1 Initialization

For the sake of completeness, we recall in this paragraph the regularization approach used in the original MuLoG approach, that we still use for the initialization of MuLoG. Under the assumption that the radar properties of the scene vary slowly with the spatial location, a spatial averaging can be used to estimate a first guess of the covariance matrix  $\Sigma_k \stackrel{\text{def}}{=} \mathbb{E}[\mathbf{v}_k \mathbf{v}_k^*]$ . At pixel  $k$ , the so-called *radar coherence* between channels  $i$  and  $j$  of the  $D$ -channels SAR image is defined by  $\rho_{k,i,j} = |[\Sigma_k]_{i,j}| / \sqrt{[\Sigma_k]_{i,i}[\Sigma_k]_{j,j}}$ , i.e., the correlation coefficient between the two channels. This coherence is notably over-estimated by the sample covariance estimator when  $L$  is small, see [46]. In the extreme case of  $L = 1$ ,  $\mathbf{C}_k = \mathbf{v}_k \mathbf{v}_k^*$  and the coherence equals 1, see Fig. 5. Some amount of smoothing is necessary to obtain more meaningful coherence values and to guarantee that the covariance matrix  $\mathbf{C}_k^{(\text{init})}$  be positive definite. The coherence is estimated by weighted averaging with Gaussian weights:

$$\hat{\rho}_{k,i,j} = \frac{|\sum_{\ell} w_{k,\ell} [\mathbf{C}_{\ell}]_{i,j}|}{\sqrt{(\sum_{\ell} w_{k,\ell} [\mathbf{C}_{\ell}]_{i,i})(\sum_{\ell} w_{k,\ell} [\mathbf{C}_{\ell}]_{j,j})}}, \quad (31)$$

where the weights  $w_{k,\ell}$  are defined, based on the spatial distance between pixels  $k$  and  $\ell$ , using a Gaussian kernel with spatial variance  $\tau/2\pi$ . Spatial parameter  $\tau$  is chosen in order to achieve a trade-off between bias reduction and resolution loss. Figure 5 shows coherence images  $\hat{\rho}$  of a single-look interferometric pair ( $L = 1$  and  $D = 2$ ) for different values of  $\tau$ . We use the empirical rule  $\tau = D / \min(L, D)$  to increase the filtering strength with the dimensionality of SAR images and reduce it for multi-look images.

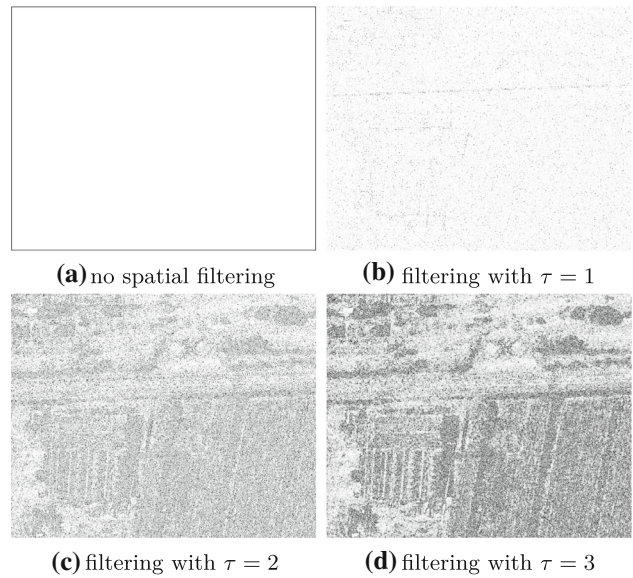


Fig. 5 Estimation of the interferometric coherence by local Gaussian filtering, for different filtering strengths  $\tau$ . In the absence of spatial filtering, the coherence is equal to 1 everywhere. All images are displayed using the same gray-level scale with a zero coherence displayed in black and a unit coherence displayed in white. The area shown corresponds to the images of Fig. 2

The estimated coherence values  $\hat{\rho}_{k,i,j}$  are then used to define a first estimate  $\hat{\mathbf{C}}_k$  of the covariance matrix at pixel  $k$ :

$$[\hat{\mathbf{C}}_k]_{i,j} = \hat{\rho}_{k,i,j} \frac{\sqrt{[\mathbf{C}_k]_{i,i}[\mathbf{C}_k]_{j,j}}}{|[\mathbf{C}_k]_{i,j}|} [\mathbf{C}_k]_{i,j}. \quad (32)$$

Note that on-diagonal entries are unchanged:  $\forall i, [\hat{\mathbf{C}}_k]_{i,i} = [\mathbf{C}_k]_{i,i}$ , while the magnitude of off-diagonal entries becomes equal to  $\hat{\rho}_{k,i,j}$  and the phase of off-diagonal entries is preserved. In practice, the coherence  $\hat{\rho}_{k,i,j}$  is smaller than 1 for all values of  $\tau$  larger than 0. In order to guarantee that the covariance matrix  $\hat{\mathbf{C}}_k$  actually is positive definite, off-diagonal entries can additionally be shrunk toward 0 by a small factor (by multiplication by a factor .99 for example). The filtering step applied in equations (31) and (32) largely improves the conditioning of covariance matrix  $\hat{\mathbf{C}}_k$ , which helps performing the principal component analysis required to define transform  $\Omega$  (by producing a more stable analysis).

In contrast to intensity data for which  $\mathbb{E}\{[\mathbf{C}_k]_{ij}\} = [\Sigma_k]_{ij}$  for all channels  $i$  and  $j$ , the average of log-transformed data is known to suffer from a systematic bias [49] that can be quantified on diagonal elements, for all  $1 \leq k \leq D$ , by

$$\mathbb{E}[\log \mathbf{C}_{kk}] - \log \Sigma_{kk} = \psi(L) - \log L, \quad (33)$$

where  $\psi$  is the di-gamma function. MuLoG does not estimate  $\log \Sigma$  by averaging, but by iterating the lines 8 to 11 in Algorithm 1. The sequence of these iterations leads to an



**Algorithm 4:** Initial estimation of the covariances

**input** : a bidimensional field of speckle-corrupted covariance matrices  $\mathbf{C}$   
**output**: the initial guess  $\mathbf{C}^{(\text{init})}$   
 (to be used line 1 of Algorithm 1)

- 1  $\tau \leftarrow D / \min(L, D)$  *(filtering strength)*
- 2  $\mathbf{C}^{(\text{filt})} \leftarrow \text{Gaussian\_filter}(\mathbf{C}, \tau / 2\pi)$   
*(filter with a Gaussian kernel of variance  $\tau / 2\pi$ )*
- 3  $\left\{ \hat{\rho}_{k,i,j} \leftarrow \frac{|[\mathbf{C}_\ell^{(\text{filt})}]_{i,j}|}{\sqrt{[\mathbf{C}_\ell^{(\text{filt})}]_{i,i}[\mathbf{C}_\ell^{(\text{filt})}]_{j,j}}} \right\}_{k=1..n, i,j=1..D}$   
*(estimate coherences)*
- 4  $\left\{ [\hat{\mathbf{C}}_k]_{i,j} \leftarrow \hat{\rho}_{k,i,j} \frac{\sqrt{[\mathbf{C}_k]_{i,i}[\mathbf{C}_k]_{j,j}}}{|[\mathbf{C}_k]_{i,j}|} [\mathbf{C}_k]_{i,j} \right\}_{k=1..n, i,j=1..D}$   
*(impose the estimated coherences)*
- 5  $\left\{ \mathbf{C}_k^{(\text{init})} \leftarrow \exp(\log L - \psi(L)) \hat{\mathbf{C}}_k \right\}_{k=1..n}$   
*(pre-compensate bias)*

unbiased estimate. However, at the first iteration, the Gaussian denoiser function  $f_\sigma$  is applied to the initial guess  $\mathbf{C}_k^{(\text{init})}$ . This denoiser performs an averaging of the values in homogeneous areas. Convergence to the final values is faster if the bias is pre-compensated in the initial guess  $\mathbf{C}^{(\text{init})}$ , as done in line 5 of Algorithm 4.

**4.2 Regularization**

The previous paragraph described the strategy to build an initial guess  $\mathbf{C}_k^{(\text{init})}$  of the covariance matrix at pixel  $k$ . This guess serves as a starting point for the iterative estimation procedure conducted by Algorithm 4. Line 2 of Algorithm 4, a regularized version  $\mathbf{C}_k^{(\text{reg})}$  of the covariance matrix is defined to compute vector  $\mathbf{y}$  (line 5) and then used to define the data-fidelity proximal operator (line 10). Although a rank-deficient matrix  $\mathbf{C}_k$  such as  $\mathbf{v}_k \mathbf{v}_k^*$  could possibly be used to define the proximal operator (by replacing  $\exp(\Omega(\mathbf{y}))$  by this rank-deficient matrix in the definition of  $F(\mathbf{x})$  in equation (10)), rank-deficient or ill-conditioned covariance matrices lead to cost functions  $F(\mathbf{x})$  that are harder to minimize. On the other hand, we show in the sequel that the spatial smoothing strategy used in Algorithm 4 to build our initial guess  $\mathbf{C}_k^{(\text{init})}$  should not be used to compute  $\mathbf{C}_k^{(\text{reg})}$  (as done in the original algorithm MuLoG [14]) since this may lead to significant biases.

In order to control the condition number of the covariance matrices  $\mathbf{C}_k^{(\text{reg})}$ , we adjust the eigenvalues by applying an affine map that rescales the eigenvalues from the range  $[\lambda_{\min}, \lambda_{\max}]$  to the range  $[\lambda_{\max}/\bar{c}, \lambda_{\max}]$ , with  $\bar{c} = \min(c, \lambda_{\max}/\lambda_{\min})$ . This transformation ensures that the

**Algorithm 5:** Regularization of covariance matrices

**input** : a covariance matrix  $\mathbf{C}_k \in \mathbb{C}^{D \times D}$   
**input** : a condition number  $c$   
**output**: a regularized matrix  $\mathbf{C}_k^{(\text{reg})} \in \mathbb{C}^{D \times D}$

- 1 compute the eigenvalue decomposition:  
 $\mathbf{C}_k = \mathbf{E} \text{diag}(\boldsymbol{\Lambda}) \mathbf{E}^*$
- 2  $\lambda_{\min} \leftarrow \min(\boldsymbol{\Lambda})$  *(smallest eigenvalue)*
- 3  $\lambda_{\max} \leftarrow \max(\boldsymbol{\Lambda})$  *(largest eigenvalue)*
- 4  $\bar{c} \leftarrow \min(c, \lambda_{\max}/\lambda_{\min})$  *(targeted condition number)*
- 5  $\{\lambda_k \leftarrow \frac{\lambda_k - \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \lambda_{\max} (1 - 1/\bar{c}) + \lambda_{\max}/\bar{c}\}_{k=1..D}$   
*(affine mapping of the eigenvalues)*
- 6  $\mathbf{C}_k^{(\text{reg})} \leftarrow \mathbf{E} \text{diag}(\boldsymbol{\Lambda}) \mathbf{E}^*$   
*(recomposition with the modified eigenvalues)*

resulting covariance matrix has a condition number at most equal to  $c$ . Moreover, the largest eigenvalue  $\lambda_{\max}$  is left unchanged and the ordering of the eigenvalues is preserved by this strictly increasing mapping (provided that  $c > 1$ ). It seems that this latter property is beneficial to limit the bias introduced by the covariance regularization scheme. If the condition number is larger than the actual condition number of  $\mathbf{C}_k$ , the affine map corresponds to the identity map. The computation of the regularized covariance matrices is summarized in Algorithm 5. We use in the following the value  $c = 10^3$ .

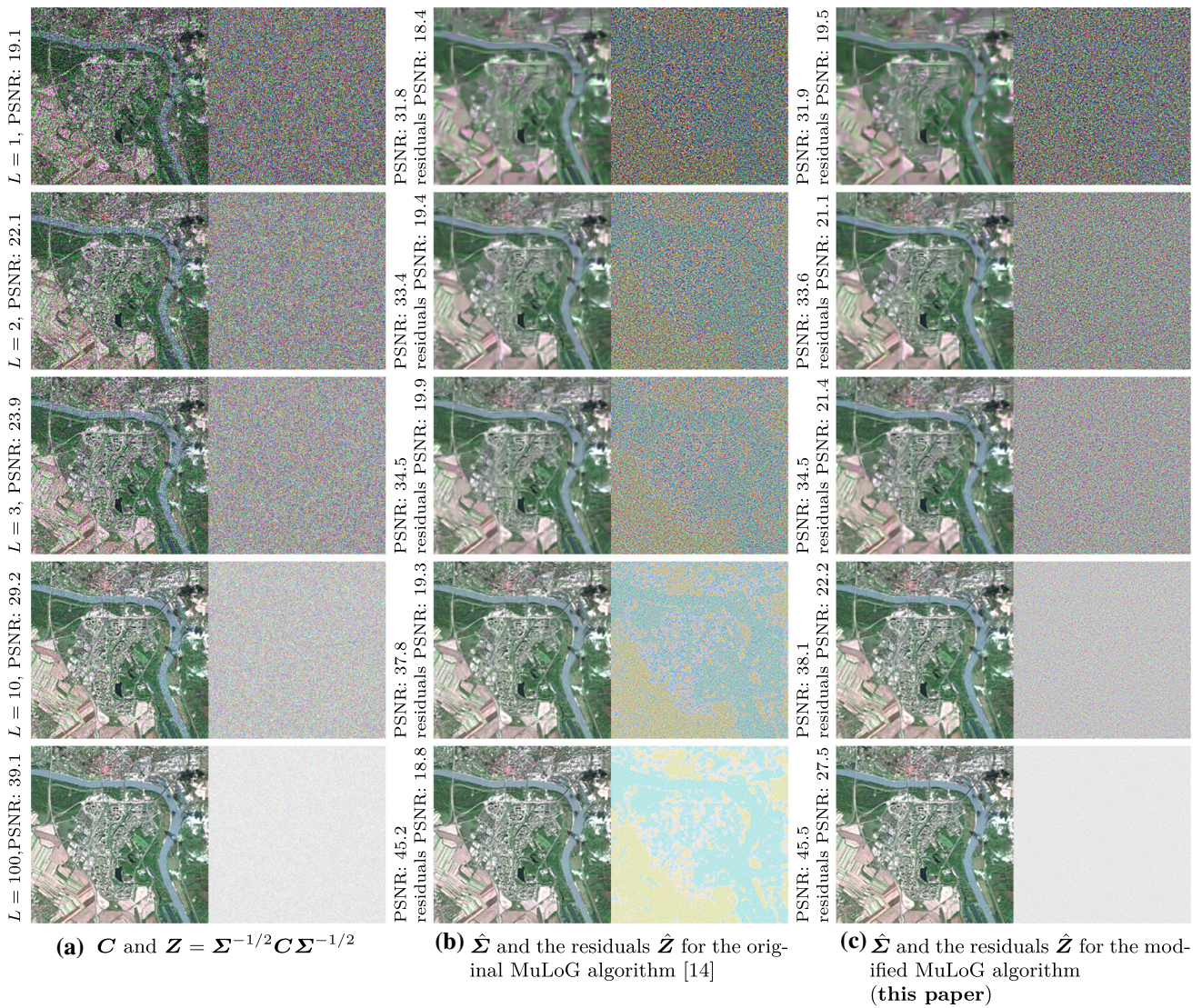
**5 Numerical Experiments**

**5.1 Simulated Data**

In a first experiment, we compare the impact of the modifications introduced in Sects. 3 and 4 with respect to the original MuLoG algorithm in [14] on a simulated PolSAR image generated from optical satellite images by building at each pixel index  $k$  the following covariance matrix

$$\boldsymbol{\Sigma}_k = \frac{1}{4} \begin{pmatrix} (G_k + R_k)^2 & 0 & (G_k^2 - R_k^2)(1 + j) \\ 0 & 4B_k^2 & 0 \\ (G_k^2 - R_k^2)(1 - j) & 0 & (G_k - R_k)^2 \end{pmatrix}, \tag{34}$$

where  $R_k, G_k$  and  $B_k$  are the red–green–blue (RGB) channels of the optical image, and the polarimetric channels of the covariance matrix are organized in the following order  $HH, HV$  and  $VV$ . This way the optical image coincides with the RGB representation of  $\boldsymbol{\Sigma}$  when represented by fake colors in the Pauli basis, as described page 3. This model considers that channel HV is decorrelated from channels HH and VV, while channels HH and VV have a correlation of  $1/\sqrt{2} \approx 0.71$ . Given such a ground-truth image  $\boldsymbol{\Sigma}$ , we next simulated noisy



**Fig. 6** Comparisons of the original MuLoG algorithm [14] and the modified version introduced in this paper, for different values of  $L$ . (a) Noisy image  $C$  and its noise component  $Z = \Sigma^{-1/2} C \Sigma^{-1/2}$  simulated from the 2003 SPOT Satellite image of Bratislava. (b-c) Estimates  $\hat{\Sigma}$  and the corresponding residuals  $\hat{Z} = \hat{\Sigma}^{-1/2} C \hat{\Sigma}^{-1/2}$  (method noise) for

the original and modified versions of MuLoG. The residuals  $\hat{Z}$  should ideally be as close as possible to the actual noise component  $Z$ , we report both the restoration PSNR ( $-10 \log_{10} \|\hat{\Sigma} - \Sigma\|_2^2$ ) and the residuals PSNR ( $-10 \log_{10} \|\hat{Z} - Z\|_2^2$ ) for each estimator. The ground truth is not displayed but is extremely close to  $C$  when  $L = 100$  (last row)

versions  $C$  with  $L$  looks by random sampling, at each pixel index  $k$ ,

$$C_k = \Sigma_k^{1/2} Z_k \Sigma_k^{1/2} \text{ with } Z_k = \frac{1}{L} \sum_{l=1}^L (v_k^{(l)})(v_k^{(l)})^*, \quad (35)$$

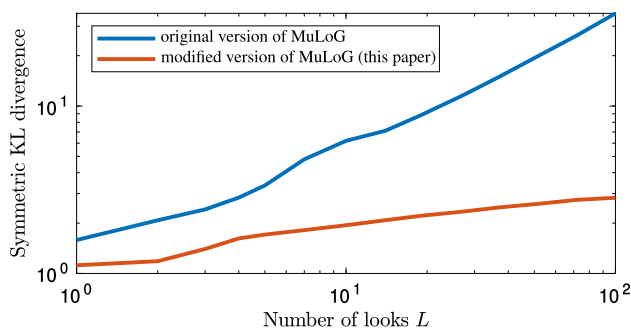
where  $\Sigma_k^{1/2}$  denotes the Hermitian square root of  $\Sigma_k^{1/2}$ , and  $v_k^{(l)}$  are independent complex random vectors with real and imaginary parts drawn according to a Gaussian white noise with standard deviation  $1/\sqrt{2}$ . By construction, this gives  $Z_k \sim \mathcal{W}(\text{Id}_D, L)$  and  $C_k \sim \mathcal{W}(\Sigma, L)$ .

### 5.2 Evaluation with Simulations

Using the procedure described in the previous paragraph, we simulated a PolSAR data  $\Sigma$  from a 2003 SPOT Satellite image of Bratislava (Slovakia)<sup>2</sup>. We suggest performing first a visual comparison of the estimated  $\hat{\Sigma}$ , respectively, obtained by the original version of MuLoG and by the modified version introduced in this paper, on noisy versions  $C$

<sup>2</sup> Provided by CNES under Creative Commons Attribution-Share Alike 3.0 Unported license. See: [https://commons.wikimedia.org/wiki/File:Bratislava\\_SPOT\\_1027.jpg](https://commons.wikimedia.org/wiki/File:Bratislava_SPOT_1027.jpg).

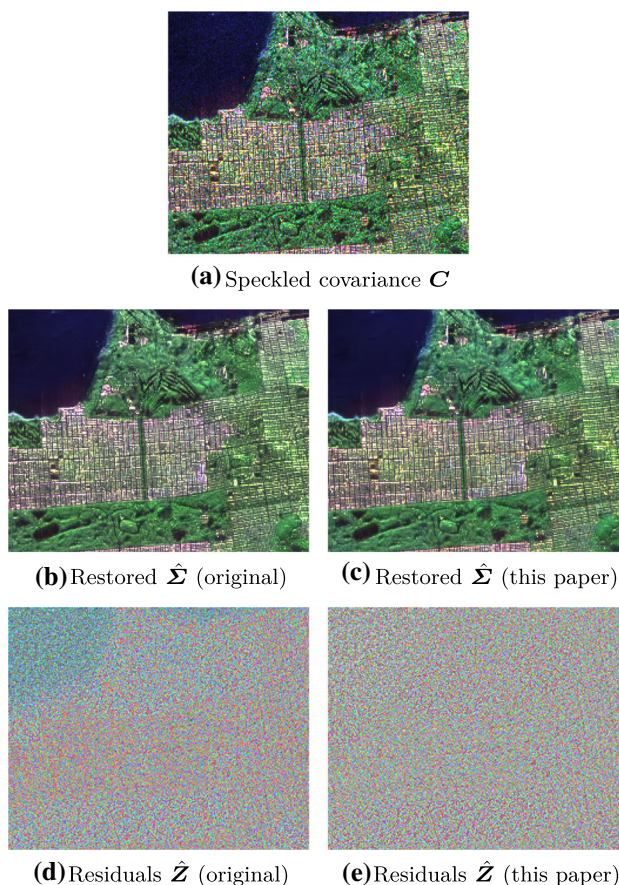




**Fig. 7** Evolution of the symmetric KL divergence between the ground truth  $\Sigma$  and the estimated residuals  $\hat{\Sigma}$  obtained with the original or the modified version of MuLoG. The divergence is computed on the simulated PolSAR data obtained from the 2003 SPOT Satellite image of Bratislava shown on Fig. 6, for various number of looks  $L$

obtained for five different numbers of looks:  $L = 1, 2, 3, 10$  and  $100$ . In addition, in order to get more insight into the behavior of each estimator, we display the residuals (aka, method noise [4]), defined by  $\hat{Z} = \hat{\Sigma}^{-1/2} C \hat{\Sigma}^{-1/2}$  where  $\hat{\Sigma}^{-1/2}$  is the inverse of the Hermitian square root  $\hat{\Sigma}^{1/2}$ . Should the estimation be perfect,  $\hat{\Sigma}$  would exactly be equal to  $\Sigma$ , and  $\hat{Z}$  would perfectly match the white speckle component  $Z$ : the residuals would be signal-independent. Comparing  $\hat{Z}$  to  $Z$  is thus an efficient way to assess the bias/variance trade-off achieved by the estimator. Areas of  $\hat{Z}$  that appear less noisy than  $Z$  indicate an under-smoothing. If  $\hat{Z}$  contains structures not present in  $Z$ , this indicates an over-smoothing. Wherever  $\hat{Z}$  has a different color than  $Z$ , this is a sign of bias.

Results are provided in Fig. 6. The estimated images  $\hat{\Sigma}$  obtained with the original version of MuLoG and the modified version introduced in this paper are visually very close. As expected, the quality of the restored images improves with the number of looks along with the signal-to-noise ratio in the speckle-corrupted images. The residuals  $\hat{Z}$  clearly indicate a bias with the original version of MuLoG. This bias is smaller with respect to the speckle variance when the number of looks  $L$  is lower. This is reflected by the PSNR values given in Fig. 6: the difference between the original and the modified versions of MuLoG is more pronounced for large values of  $L$ . When the number of looks  $L$  is larger, the speckle fluctuations are smaller and the PSNR of the speckled image is higher (see Fig. 6a). Unsurprisingly, the PSNR of the restored images gradually increases with the PSNR of the speckled image for both versions of MuLoG (see Fig. 6b, c). Computing the PSNR of the residuals  $\hat{Z}$  of the original MuLoG algorithm shows that the residuals are affected by an error that is increasingly large with respect to the level of fluctuations of the residuals, due to the presence of a bias. The modified MuLoG algorithm introduced in this paper, on the other hand, leads to residuals closer and closer to the true residuals when  $L$  increases.



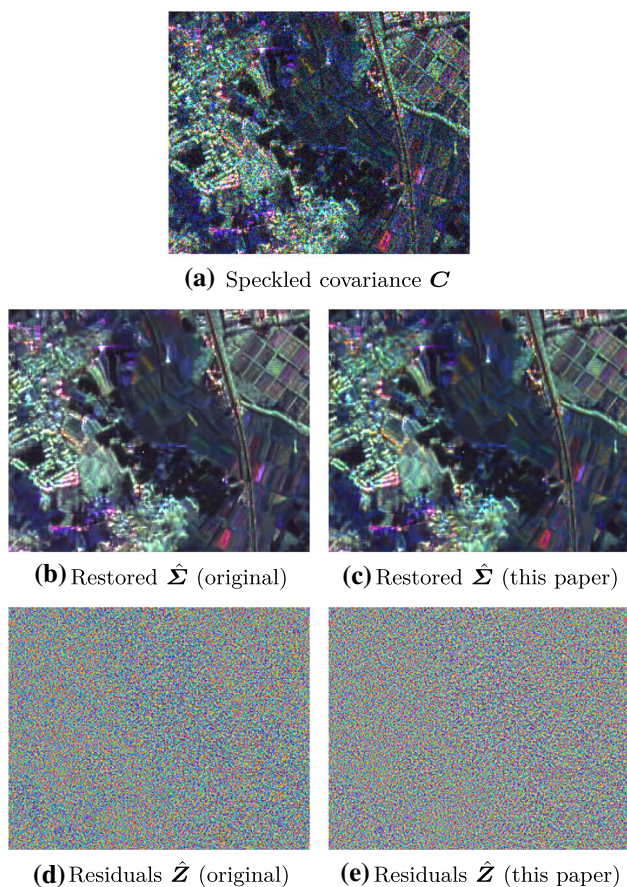
**Fig. 8** Despeckling performance comparison of the original algorithm MuLoG and the modified version introduced in this paper. AIRSAR polarimetric image of San-Francisco, USA (©NASA): **a** The original image; **b, c** the despeckling results; **d–e** the residuals

The residuals  $\hat{Z}$  obtained with the modified MuLoG algorithm are comparable to the true residuals  $Z$ : no geometrical structure from the image is noticeable in the residuals, which indicates that contrasted features were not removed from the image by the despeckling processing. The levels of fluctuations in  $\hat{Z}$  and  $Z$  seem similar. In order to perform a more quantitative comparison of the residuals, we report in Fig. 7 the symmetrical Kullback–Leibler divergence (KLD) between the distribution of the residuals  $\Sigma$  and  $\hat{\Sigma}$  for different numbers of looks ranging from  $L = 1$  to  $100$ . The KLD, averaged over the pixel index  $k$ , between the distributions  $\mathcal{W}(\Sigma_k; L)$  and  $\mathcal{W}(\hat{\Sigma}_k; L)$  is defined by

$$SD_{\mathcal{W}}(\hat{\Sigma}_k \| \Sigma_k) = L \text{tr}(\Sigma_k \hat{\Sigma}_k^{-1} + \Sigma_k^{-1} \hat{\Sigma}_k) - 2LD. \quad (36)$$

A KLD of 0 indicates a perfect match, while a larger value indicates a discrepancy. The divergence increases with the number of looks, which is expected because the KLD is a measure of divergence relative to the signal-to-noise ratio: the larger the signal-to-noise ratio, the more conserving is the KLD. We observe that for all values of  $L$ , the modified





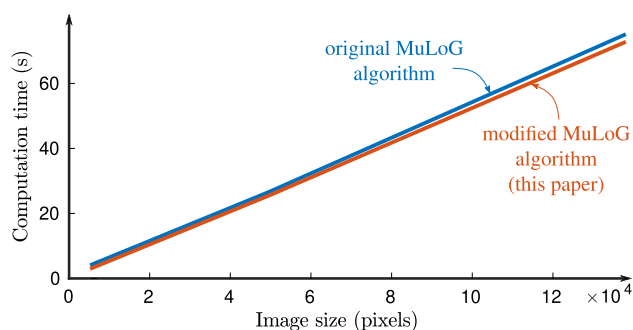
**Fig. 9** Despeckling performance comparison of the original algorithm MuLoG and the modified version introduced in this paper. PISAR polarimetric image of Tsukuba, Japan (©JAXA): **a** The original image; **b, c** the despeckling results; **d, e** the residuals

version of MuLoG leads to residuals closer to the theoretical distribution of speckle residuals. This is in agreement with the behavior observed on Fig. 6 where the bias is seen to become prominent for large numbers of looks with the original version of MuLoG.

We checked by comparing the average running time on several images that the modifications introduced in this paper do not slow down MuLoG: a slight speedup was even observed in our experiments.

### 5.3 Evaluation with Real Data

Figures 8, 9 and 11 compare the restoration performance of the original MuLoG algorithm and of the modified version introduced in this paper on PolSAR images from 3 different sensors (AIRSAR from NASA, PISAR from JAXA and SETHI from ONERA). The equivalent numbers of looks, estimated by maximum likelihood, are, respectively, equal to 2.7, 1.7 and 1 for each image. From the 1 look image of SETHI, we build a 4 looks image by spatial averaging and



**Fig. 10** Evolution of the mean computation time (in seconds) with respect to the number of pixels  $n$  of a PolSAR image ( $D = 3$ ) obtained, respectively, by MuLoG v1 and MuLoG v2 when embedding BM3D

downsampling by a factor 2 in the horizontal and vertical directions.

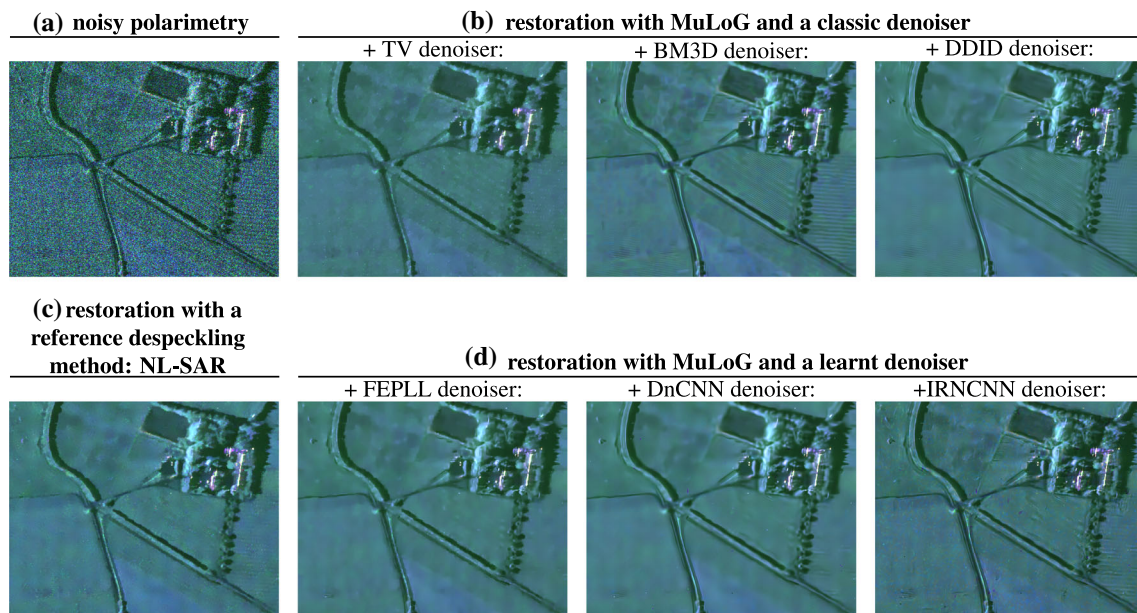
As previously observed on simulated data, while the results of the two versions of the algorithm are similar when the number of looks is small, a bias is visible in the residuals with the original MuLoG algorithm for larger equivalent numbers of looks.

### 5.4 Computation Time Analysis

Figure 10 compares the computation time as a function of the number of pixels of the original MuLoG algorithm and the modified version introduced in this paper (the denoising step is performed for both algorithms by BM3D). Computation times were averaged over 20 executions for both algorithms. Each algorithm runs the same fixed number of ADMM iterations and quasi-Newton iterations. As expected, the computation time grows linearly with the number of pixels. Despite the refined computations of the first-order and second-order derivatives introduced in this paper, the computation time per iteration is not increased: there is even a slight acceleration of the method (Fig. 11).

## 6 Discussion

MuLoG is a generic framework that offers the possibility of a straightforward application of denoisers developed for additive white Gaussian noise (i.e., optical imaging). It suppresses the need for a time-consuming adaptation of these algorithms to the specifics of SAR imagery. Beyond a much faster transfer of state-of-the-art denoising methods, it makes it easier to run several denoisers in parallel and compare the restored images obtained by each. Figure 12 illustrates such restoration results obtained with 6 different denoising techniques. The patch-based despeckling method NL-SAR [17] is also applied to serve as a reference computed without using the MuLoG framework. The images produced with MuLoG have



**Fig. 12** Many different Gaussian denoisers can be used in conjunction with MuLoG. A speckled polarimetric image (a) is restored with MuLoG and conventional Gaussian denoisers (b) (total variation minimization [42], the block-matching collaborative 3D filtering algorithm BM3D [10], the dual domain image denoising method DDID [28]) or

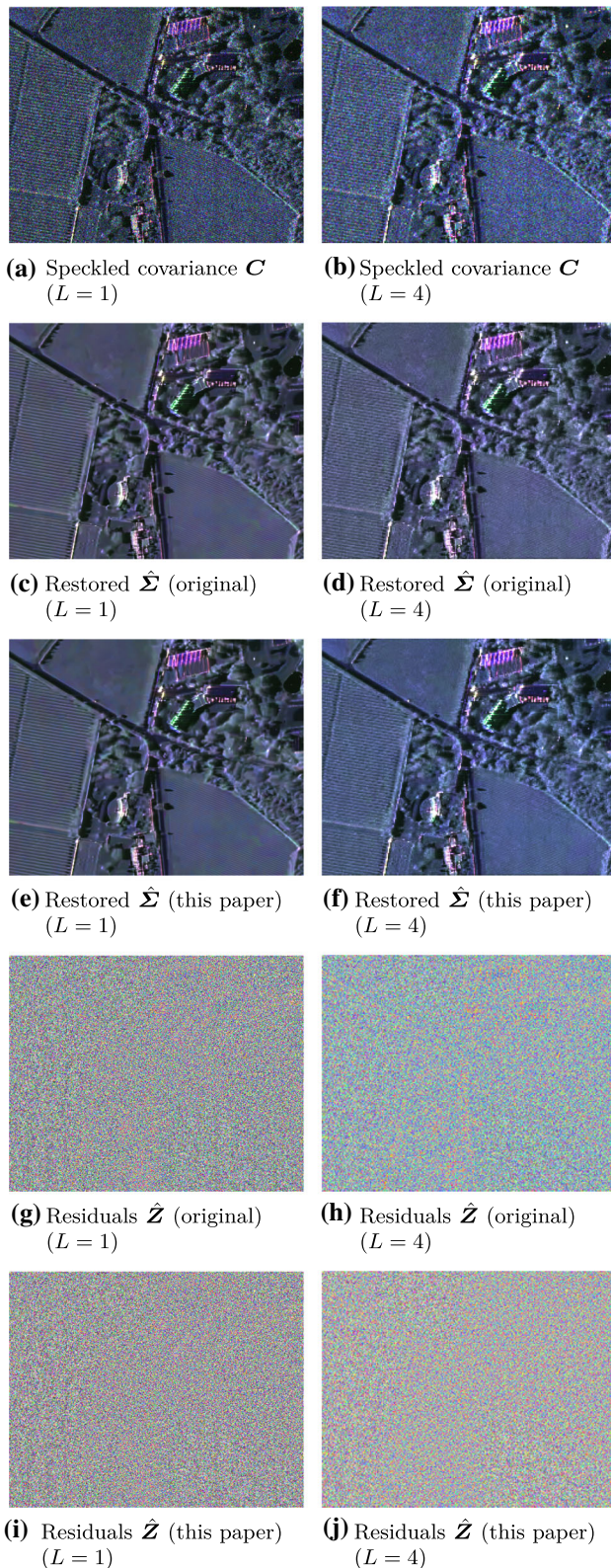
more recent learning-based techniques (d) (a fast algorithm based on a model of image patches as a mixture of Gaussians FEPLL [37]; and two deep neural networks: DnCNN [50] and IRNCNN [51]). The restored image obtained with the patch-based despeckling method NL-SAR [17] is shown as a reference (restoration not based on MuLoG)

a quality that is on-par with NL-SAR, with some denoising algorithms better at restoring textures, others at smoothing homogeneous areas or at preserving the continuity of the roads (dark linear structures). Method-specific artifacts can be identified in the results: the total-variation denoiser [42] suppresses oscillating patterns and creates artificial edges in smoothly varying regions; BM3D [10], based on wavelets transforms, is very good at restoring oscillating patterns but it may also produce oscillating artifacts in some low signal-to-noise ratio areas; DDID [28] creates oscillating artifacts and some ripples around edges; FEPLL [37] tends to suppress low signal-to-noise ratio oscillating patterns; DnCNN [50] produces smooth images with some point-like artifacts and a suppression of the low signal-to-noise ratio oscillating patterns; IRNCNN [51] better restores some of the oscillating patterns but introduces many artifacts in the form of fake-edges in homogeneous areas. We believe that the possibility to include deep learning techniques with MuLoG is particularly useful to multi-dimensional SAR imaging for which the supervised training of dedicated networks is very difficult to achieve due to the lack of ground-truth data and the dimensionality of the patterns (spatial patterns that extend in the  $D^2$  real-valued dimensions of the covariance matrices).

One may wonder to what extent Gaussian denoisers developed to remove white Gaussian noise also behave satisfyingly to handle the non-Gaussian fluctuations that appear throughout the iterations in MuLoG’s transformed domain.

A possible explanation of the success of the plug-in ADMM approach, illustrated in Fig. 12 with several denoisers including deep neural networks trained specifically for Gaussian noise, is that throughout the iterations a trade-off between data fidelity and regularity is found by repeatedly applying the Gaussian denoiser to project onto the manifold of smooth images. This interpretation indicates that the Gaussian denoiser plays a role somewhat different to its original role. Given the specific nature of both the non-Gaussian fluctuations that appear throughout MuLoG’s iterations and the specificity of SAR scenes, learning dedicated denoisers (for example in the form of deep neural networks) is a natural way to further improve the despeckling algorithm. Since the distribution of fluctuations varies from one iteration to the next, specific denoisers should be used at each iteration: this corresponds to the “algorithm unrolling” approach [35] discussed below with other possible extensions of our work. In the channels shown in Fig. 3, one can observe linear structures, edges and textures that are also common to natural images. It is therefore not unreasonable to think that a denoiser suitable to denoise natural images can also be applied to the log-transformed channels. Figure 12 supports this: very different Gaussian denoisers can be applied within the MuLoG framework and provide decent restoration results. By using a denoiser specific to the content of SAR scenes, the restoration performance can be slightly improved, but a recent study [13] has shown that, for the restoration of intensity SAR





**Fig. 11** Despeckling performance comparison of the original algorithm MuLoG and the modified version introduced in this paper. SETHI polarimetric image of Nîmes area, France (©ONERA): first column, single-look image; second column, 4-looks image. **a, b** The original images; **c–f** the despeckling results; **g–j** the residuals

images with MuLoG, a deep neural network trained on natural images behaves almost as well as the same network trained specifically on SAR images. While single-channel speckle-free SAR images can be obtained by averaging long time-series of images and subsequently used for the supervised training of despeckling networks, this approach is hardly feasible in SAR polarimetry or interferometry due to the lack of data. Progress in self-supervised training strategies for remote sensing data restoration is needed to enable the learning of more specific networks [11,39].

This paper introduced several modifications to the MuLoG algorithm [14] based on the closed-form derivation of first and second-order derivatives of data-fidelity proximal operator. These mathematical developments can benefit other algorithms beyond MuLoG. Since the introduction of the plug-in-ADMM strategy in [48], there have been numerous works which considered extensions to proximal algorithms [33], half quadratic splitting [51] or formulations to include a Gaussian denoiser as an implicit regularizer [41]. The application of these approaches to multi-channel SAR data requires the evaluation of the data-fidelity proximal operator described in Sect. 3 or at least the evaluation of the gradient of the data-fidelity term (Algorithm 2). Another rapidly growing body of works considers the end-to-end training of unrolled algorithms, where deep neural networks are trained to remove noise and artifacts while gradient descent terms derived from the data-fidelity term are interlaced between two applications of neural networks [35]. The computation of the exact gradient by Algorithm 2 is then essential to apply these techniques to multi-channel SAR images. Finally, several recent works [24,26,27,29] consider Monte Carlo Markov chains to sample images according to a posterior distribution. These approaches combine a gradient step according to the data-fidelity term, a regularization step computed with a black-box denoiser (similarly to the “regularization by denoising” (RED) approach [41]), and a random perturbation (Langevin dynamics). The exact gradient of the data-fidelity term given in Algorithm 2 is once again necessary for the application to SAR imaging.

**Acknowledgements** The airborne SAR images processed in this paper were provided by ONERA, the French aerospace lab, within the project ALYS ANR-15-ASTR-0002 funded by the DGA (Direction Générale à l’Armement) and the ANR (Agence Nationale de la Recherche). This work has been supported in part by the French space agency CNES under project R-S19/OT-0003-086.

## A Gradient of the Objective $F$

The gradient of (10) is given by:

$$\mathbf{g} = \beta(\mathbf{x} - \mathbf{z}) + L\Theta\left(\text{Id}_D - \frac{\partial e^{\Sigma}}{\partial \Sigma}\Big|_{-\Omega(\mathbf{x})}^* [e^{\Omega(\mathbf{y})}]\right) \quad (12)$$



**Proof** (Proof of eq. (12)) Applying the chain rule to eq. (10) leads to the following decomposition

$$\begin{aligned}
 \mathbf{g} &= \nabla_{\mathbf{x}} \left[ \frac{\beta}{2} \|\mathbf{x} - \mathbf{z}\|^2 + L \operatorname{tr}(\Omega(\mathbf{x}) + e^{\Omega(\mathbf{y})} e^{-\Omega(\mathbf{x})}) \right] \quad (37) \\
 &= \beta(\mathbf{x} - \mathbf{z}) + L \left. \frac{\partial \Omega(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}}^* \left( \nabla_{\mathbf{X}} \operatorname{tr} \mathbf{X} \Big|_{\Omega(\mathbf{x})} \right. \\
 &\quad \left. - \frac{\partial e^{\mathbf{X}}}{\partial \mathbf{X}} \Big|_{-\Omega(\mathbf{x})}^* \frac{\partial e^{\Omega(\mathbf{y})} \mathbf{X}}{\partial \mathbf{X}} \Big|_{e^{-\Omega(\mathbf{x})}}^* \nabla_{\mathbf{X}} \operatorname{tr} \mathbf{X} \Big|_{e^{\Omega(\mathbf{y})} e^{-\Omega(\mathbf{x})}} \right). \quad (38)
 \end{aligned}$$

Using that for any matrix  $\mathbf{A}$  and  $\mathbf{B}$

$$\left. \frac{\partial \mathbf{A} \mathbf{X}}{\partial \mathbf{X}} \right|_{e^{-\Omega(\mathbf{x})}}^* [\mathbf{B}] = \mathbf{B} \mathbf{A}^*, \quad (39)$$

$$\left. \frac{\partial \Omega(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}}^* [\mathbf{B}] = \Theta(\mathbf{B}), \quad (40)$$

$$\nabla_{\mathbf{X}} \operatorname{tr} \mathbf{X} \Big|_{e^{\Omega(\mathbf{y})} e^{-\Omega(\mathbf{x})}} = \operatorname{Id}_D. \quad (41)$$

concludes the proof.  $\square$

### B Proof of Proposition 1

**Proposition 1** Let  $\Sigma$  be a  $D \times D$  Hermitian matrix with distinct eigenvalues. Let  $\Sigma = \mathbf{E} \operatorname{diag}(\Lambda) \mathbf{E}^*$  be its eigendecomposition where  $\mathbf{E}$  is a unitary matrix of eigenvectors and  $\Lambda = (\lambda_1, \dots, \lambda_D)$  the vector of corresponding eigenvalues. Then, for any  $D \times D$  Hermitian matrix  $\mathbf{A}$ , denoting  $\bar{\mathbf{A}} = \mathbf{E}^* \mathbf{A} \mathbf{E}$ , we have

$$\left. \frac{\partial e^{\Sigma}}{\partial \Sigma} \right|_{\Sigma} [\mathbf{A}] = \mathbf{E} [\mathbf{G} \odot \bar{\mathbf{A}}] \mathbf{E}^* \quad (17)$$

where  $\odot$  is the element-wise (a.k.a, Hadamar) product, and, for all  $1 \leq i, j \leq D$ , we have defined

$$\mathbf{G}_{i,j} = \begin{cases} \frac{e^{\lambda_i} - e^{\lambda_j}}{\lambda_i - \lambda_j} & \text{if } i \neq j, \\ e^{\lambda_i} & \text{otherwise.} \end{cases} \quad (18)$$

**Proof** Let us start by recalling the following Lemma whose proof can be found in [5,18,31].  $\square$

**Lemma 1** Let  $\Sigma$  be an Hermitian matrix with distinct eigenvalues. Let  $\Sigma = \mathbf{E} \operatorname{diag}(\Lambda) \mathbf{E}^*$  be its eigendecomposition where  $\mathbf{E}$  is a unitary matrix of eigenvectors and  $\Lambda = (\lambda_1, \dots, \lambda_D)$  the vector of corresponding eigenvalues. We have for a Hermitian matrix  $\mathbf{A}$

$$\frac{\partial \Lambda}{\partial \Sigma} [\mathbf{A}] = \operatorname{diag}(\bar{\mathbf{A}}) \quad \text{and} \quad \frac{\partial \mathbf{E}}{\partial \Sigma} [\mathbf{A}] = \mathbf{E} (\mathbf{J} \odot \bar{\mathbf{A}}) \quad (42)$$

where  $\bar{\mathbf{A}} = \mathbf{E}^* \mathbf{A} \mathbf{E}$  and  $\mathbf{J}$  is the skew-symmetric matrix

$$\mathbf{J}_{ij} = \begin{cases} \frac{1}{\lambda_j - \lambda_i} & \text{if } i \neq j, \\ 0 & \text{otherwise.} \end{cases} \quad (43)$$

Recall that  $\exp \Sigma = \mathbf{E} \operatorname{diag}(e^{\Lambda}) \mathbf{E}^*$ . From Lemma 1, by applying chain rule, we have

$$\frac{\partial e^{\Sigma}}{\partial \Sigma} [\mathbf{A}] = \mathbf{E} (\mathbf{J} \odot \bar{\mathbf{A}}) \operatorname{diag}(e^{\Lambda}) \mathbf{E}^* \quad (44)$$

$$+ \mathbf{E} \operatorname{diag}(e^{\Lambda}) (\mathbf{J} \odot \bar{\mathbf{A}})^* \mathbf{E}^* \quad (45)$$

$$+ \mathbf{E} \operatorname{diag}(e^{\Lambda}) \operatorname{diag}(\bar{\mathbf{A}}) \mathbf{E}^*. \quad (46)$$

We have for  $i \neq j$

$$\left[ (\mathbf{J} \odot \bar{\mathbf{A}}) \operatorname{diag}(e^{\Lambda}) + \operatorname{diag}(e^{\Lambda}) (\mathbf{J} \odot \bar{\mathbf{A}})^* \right]_{ij} \quad (47)$$

$$= \mathbf{J}_{ij} \bar{\mathbf{A}}_{ij} e^{\lambda_j} + e^{\lambda_i} \mathbf{J}_{ji} \bar{\mathbf{A}}_{ji} = \mathbf{J}_{ij} \bar{\mathbf{A}}_{ij} e^{\lambda_j} - e^{\lambda_i} \mathbf{J}_{ij} \bar{\mathbf{A}}_{ij} \quad (48)$$

$$= \mathbf{J}_{ij} \bar{\mathbf{A}}_{ij} (e^{\lambda_j} - e^{\lambda_i}) = \bar{\mathbf{A}}_{ij} \mathbf{G}_{ij} = [\mathbf{G} \odot \bar{\mathbf{A}}]_{ij}. \quad (49)$$

For  $i = j$ , since  $\mathbf{J}_{ii} = 0$ , we conclude the proof.  $\square$

### C Proof of Corollary 1

**Corollary 1** Let  $\Sigma$  be a Hermitian matrix with distinct eigenvalues. The Jacobian of the matrix exponential is a self-adjoint operator

$$\left. \frac{\partial e^{\Sigma}}{\partial \Sigma} \right|_{\Sigma}^* = \left. \frac{\partial e^{\Sigma}}{\partial \Sigma} \right|_{\Sigma}. \quad (20)$$

**Proof** We need to prove that for any two  $D \times D$  Hermitian matrices  $\mathbf{A}$  and  $\mathbf{B}$ , we have

$$\left\langle \left. \frac{\partial e^{\Sigma}}{\partial \Sigma} [\mathbf{A}], \mathbf{B} \right\rangle = \left\langle \mathbf{A}, \left. \frac{\partial e^{\Sigma}}{\partial \Sigma} [\mathbf{B}] \right\rangle \quad (50)$$

for the matrix dot product  $\langle \mathbf{X}, \mathbf{Y} \rangle = \operatorname{tr}[\mathbf{X} \mathbf{Y}^*]$ . According to Proposition 1, this amounts to show

$$\operatorname{tr}(\mathbf{E} [\mathbf{G} \odot (\mathbf{E}^* \mathbf{A} \mathbf{E})] \mathbf{E}^* \mathbf{B}^*) = \operatorname{tr}(\mathbf{E} [\mathbf{G} \odot (\mathbf{E}^* \mathbf{B} \mathbf{E})] \mathbf{E}^* \mathbf{A}^*) \quad (51)$$

where  $\mathbf{E}$  and  $\mathbf{G}$  are defined from  $\Sigma$  as in Proposition 1. Denoting  $\bar{\mathbf{A}} = \mathbf{E}^* \mathbf{A} \mathbf{E}$  and  $\bar{\mathbf{B}} = \mathbf{E}^* \mathbf{B} \mathbf{E}$ , this can be recast as

$$\operatorname{tr}((\mathbf{G} \odot \bar{\mathbf{A}}) \bar{\mathbf{B}}^*) = \operatorname{tr}((\mathbf{G} \odot \bar{\mathbf{B}}) \bar{\mathbf{A}}^*) \quad (52)$$

Expanding the left hand side allows us to conclude the proof as follows

$$\text{tr}((\mathbf{G} \odot \bar{\mathbf{A}})\bar{\mathbf{B}}^*) = \sum_{k=1}^D \sum_{l=1}^D (\mathbf{G} \odot \bar{\mathbf{A}})_{kl} (\bar{\mathbf{B}}^*)_{lk} \tag{53}$$

$$= \sum_{k=1}^D \sum_{l=1}^D \mathbf{G}_{kl} \bar{\mathbf{A}}_{kl} \bar{\mathbf{B}}_{kl} \tag{54}$$

$$= \sum_{k=1}^D \sum_{l=1}^D \mathbf{G}_{kl} \bar{\mathbf{B}}_{kl} (\bar{\mathbf{A}})^*_{lk} \tag{55}$$

$$= \text{tr}((\mathbf{G} \odot \bar{\mathbf{B}})\bar{\mathbf{A}}^*) . \tag{56}$$

### D Hessian of the Objective F

The second-order derivative used in the approximation of the Hessian given in (23) takes the form:

$$\gamma = \beta + L \left\langle \Theta^*(\mathbf{d}), \frac{\partial^2 e^\Sigma}{\partial \Sigma^2} \Big|_{-\Omega(\mathbf{x})} [e^{\Omega(\mathbf{y})}, \Theta^*(\mathbf{d})] \right\rangle . \tag{24}$$

**Proof** (Proof of eq. (24)) Applying the chain rule to eq. (12) leads to

$$\mathbf{H} = \frac{\partial}{\partial \mathbf{x}} \left[ \beta(\mathbf{x} - \mathbf{z}) + L\Theta \left( \text{Id}_D - \frac{\partial e^\Sigma}{\partial \Sigma} \Big|_{-\Omega(\mathbf{x})}^* [e^{\Omega(\mathbf{y})}] \right) \right] \tag{57}$$

$$= \beta \text{Id}_D - L\Theta \left( \frac{\partial}{\partial \mathbf{x}} \left( \frac{\partial e^\Sigma}{\partial \Sigma} \Big|_{-\Omega(\mathbf{x})}^* [e^{\Omega(\mathbf{y})}] \right) [\cdot] \right) \tag{58}$$

$$= \beta \text{Id}_D - L\Theta \left( \frac{\partial^2 e^\Sigma}{\partial \Sigma^2} \Big|_{-\Omega(\mathbf{x})} [e^{\Omega(\mathbf{y})}, \frac{\partial -\Omega(\mathbf{x})}{\partial \mathbf{x}} [\cdot]] \right) \tag{59}$$

$$= \beta \text{Id}_D + L\Theta \left( \frac{\partial^2 e^\Sigma}{\partial \Sigma^2} \Big|_{-\Omega(\mathbf{x})} [e^{\Omega(\mathbf{y})}, \Theta^*[\cdot]] \right) . \tag{60}$$

It follows that

$$\mathbf{H}\mathbf{d} = \beta\mathbf{d} + L\Theta \left( \frac{\partial^2 e^\Sigma}{\partial \Sigma^2} \Big|_{-\Omega(\mathbf{x})} [e^{\Omega(\mathbf{y})}, \Theta^*[\mathbf{d}]] \right) \tag{61}$$

and thus, as  $\|\mathbf{d}\|_2 = 1$ , it follows that

$$\mathbf{d}^* \mathbf{H}\mathbf{d} = \beta + L\mathbf{d}^* \Theta \left( \frac{\partial^2 e^\Sigma}{\partial \Sigma^2} \Big|_{-\Omega(\mathbf{x})} [e^{\Omega(\mathbf{y})}, \Theta^*[\mathbf{d}]] \right) \tag{62}$$

$$= \beta + L \left\langle \mathbf{d}, \Theta \left( \frac{\partial^2 e^\Sigma}{\partial \Sigma^2} \Big|_{-\Omega(\mathbf{x})} [e^{\Omega(\mathbf{y})}, \Theta^*[\mathbf{d}]] \right) \right\rangle \tag{63}$$

$$= \beta + L \left\langle \Theta^*[\mathbf{d}], \frac{\partial^2 e^\Sigma}{\partial \Sigma^2} \Big|_{-\Omega(\mathbf{x})} [e^{\Omega(\mathbf{y})}, \Theta^*[\mathbf{d}]] \right\rangle \tag{64}$$

which concludes the proof.  $\square$

### E Proof of Proposition 2

**Proposition 2** Let  $\Sigma$  be a  $D \times D$  Hermitian matrix with distinct eigenvalues. Let  $\Sigma = \mathbf{E} \text{diag}(\boldsymbol{\Lambda}) \mathbf{E}^*$  be its eigen-decomposition where  $\mathbf{E}$  is a unitary matrix of eigenvectors and  $\boldsymbol{\Lambda} = (\lambda_1, \dots, \lambda_D)$  the vector of corresponding eigenvalues. For any  $D \times D$  Hermitian matrices  $\mathbf{A}$  and  $\mathbf{B}$ , denoting  $\bar{\mathbf{A}} = \mathbf{E}^* \mathbf{A} \mathbf{E}$  and  $\bar{\mathbf{B}} = \mathbf{E}^* \mathbf{B} \mathbf{E}$ , we have

$$\frac{\partial^2 e^\Sigma}{\partial \Sigma^2} [\mathbf{A}, \mathbf{B}] = \mathbf{E} [\mathbf{F}(\bar{\mathbf{A}}, \bar{\mathbf{B}})] \mathbf{E}^* \tag{26}$$

where, for all  $1 \leq i, j \leq D$ , we have

$$\mathbf{F}(\bar{\mathbf{A}}, \bar{\mathbf{B}})_{i,j} = \sum_{k=1}^D \phi_{i,j,k} (\bar{\mathbf{A}}_{ik} \bar{\mathbf{B}}_{jk}^* + \bar{\mathbf{B}}_{ik} \bar{\mathbf{A}}_{jk}^*) \tag{27}$$

$$\text{with } \phi_{i,j,k} = \begin{cases} \frac{\mathbf{G}_{ik} - \mathbf{G}_{jk}}{\lambda_i - \lambda_j} & \text{if } i \neq j, \\ \frac{\mathbf{G}_{ii} - \mathbf{G}_{ik}}{\lambda_i - \lambda_k} & \text{if } i = j \text{ and } k \neq i, \\ \frac{\mathbf{G}_{ii}}{2} & \text{if } i = j = k. \end{cases} \tag{28}$$

**Proof** (Proof of Proposition 2) The second directional derivative can be defined from the adjoint of the directional derivative as

$$\frac{\partial^2 e^\Sigma}{\partial \Sigma^2} [\mathbf{A}, \mathbf{B}] = \frac{\partial}{\partial \Sigma} \left( \frac{\partial e^\Sigma}{\partial \Sigma} \Big|_{\Sigma}^* [\mathbf{A}] \right) [\mathbf{B}] . \tag{65}$$

By virtue of Corollary 1, we have  $\frac{\partial e^\Sigma}{\partial \Sigma} \Big|_{\Sigma}^* = \frac{\partial e^\Sigma}{\partial \Sigma} \Big|_{\Sigma}$ , and then from Proposition 1 it follows that

$$\frac{\partial^2 e^\Sigma}{\partial \Sigma^2} [\mathbf{A}, \mathbf{B}] = \frac{\partial}{\partial \Sigma} (\mathbf{E} [\mathbf{G} \odot \bar{\mathbf{A}}] \mathbf{E}^*) [\mathbf{B}] . \tag{66}$$

In order to apply the chain rule on  $\mathbf{E} [\mathbf{G} \odot \bar{\mathbf{A}}] \mathbf{E}^*$ , let us first rewrite  $\mathbf{J}$  and  $\mathbf{G}$  in Lemma 1 and Proposition 1 as

$$\mathbf{J} = (\mathbf{1}_D \mathbf{1}_D^* - \text{Id}_D) \oslash (\text{Id}_D + \mathbf{1}_D \boldsymbol{\Lambda}^* - \boldsymbol{\Lambda} \mathbf{1}_D^*) \tag{67}$$

$$\mathbf{G} = \text{diag}(e^\Lambda) - (e^\Lambda \mathbf{1}_D^* - \mathbf{1}_D e^\Lambda) \odot \mathbf{J} \tag{68}$$

where  $\mathbf{1}_D$  is a  $D$  dimensional column vector of ones,  $\oslash$  denotes the element-wise division and  $e^\Lambda$  must be understood as the row vector  $(e^\Lambda)^*$ . From Lemma 1, we have for

a Hermitian matrix  $B$

$$\frac{\partial}{\partial \Sigma} \text{diag}(e^\Lambda) [B] = \text{diag}(e^\Lambda) \odot \bar{B} \tag{69}$$

$$\frac{\partial}{\partial \Sigma} (e^\Lambda \mathbf{1}_D^* - \mathbf{1}_D e^{\Lambda^*}) [B] \tag{70}$$

$$= \text{diag}(e^\Lambda) \text{diag}(\bar{B}) \mathbf{1}_D^* - \mathbf{1}_D \text{diag}(\bar{B})^* \text{diag}(e^\Lambda)$$

$$\frac{\partial}{\partial \Sigma} (\text{Id}_D + \Lambda \mathbf{1}_D^* - \mathbf{1}_D \Lambda^*) [B] = \text{diag}(\bar{B}) \mathbf{1}_D^* - \mathbf{1}_D \text{diag}(\bar{B})^* \tag{71}$$

where  $\bar{B} = E^* B E$ . By application of the chain rule, we get

$$\frac{\partial J}{\partial \Sigma} [B] = -(\text{diag}(\bar{B}) \mathbf{1}_D^* - \mathbf{1}_D \text{diag}(\bar{B})^*) \odot J \odot J \tag{72}$$

$$\frac{\partial G}{\partial \Sigma} [B] = \text{diag}(e^\Lambda) \odot \bar{B} \tag{73}$$

$$- [\text{diag}(e^\Lambda) \text{diag}(\bar{B}) \mathbf{1}_D^* - \mathbf{1}_D \text{diag}(\bar{B})^* \text{diag}(e^\Lambda) - G \odot (\text{diag}(\bar{B}) \mathbf{1}_D^* - \mathbf{1}_D \text{diag}(\bar{B})^*)] \odot J$$

where we used that  $(e^\Lambda \mathbf{1}_D^* - \mathbf{1}_D e^{\Lambda^*}) \odot J = -G \odot J$ . Let  $A$  be a Hermitian matrix and  $\bar{A} = E^* A E$ , by Lemma 1, we have

$$\frac{\partial \bar{A}}{\partial \Sigma} [B] = \bar{A} (J \odot \bar{B}) - (J \odot \bar{B}) \bar{A}. \tag{74}$$

We are now equipped to apply the chain rule to  $E[G \odot \bar{A}]E^*$  in the direction  $B$ , which leads us to

$$\frac{\partial^2 e^\Sigma}{\partial \Sigma^2} [A, B] = E[F(\bar{A}, \bar{B})]E^* \tag{75}$$

with  $F(\bar{A}, \bar{B}) = F_1 + F_2 + F_3$   $\tag{76}$

and  $F_1 = (J \odot \bar{B})(G \odot \bar{A}) - (G \odot \bar{A})(J \odot \bar{B})$   $\tag{77}$

$$F_2 = G \odot [\bar{A} (J \odot \bar{B}) - (J \odot \bar{B}) \bar{A}] \tag{78}$$

$$F_3 = \frac{\partial G}{\partial \Sigma} [B] \odot \bar{A}. \tag{79}$$

We have for all  $1 \leq i \leq D$  and  $1 \leq j \leq D$

$$[F_1]_{ij} = \sum_{k=1}^D J_{jk} G_{ik} \bar{A}_{ik} \bar{B}_{jk}^* + J_{ik} G_{jk} \bar{B}_{ik} \bar{A}_{jk}^* \tag{80}$$

and  $[F_2]_{ij} = -G_{ij} \sum_{k=1}^D J_{jk} \bar{A}_{ik} \bar{B}_{jk}^* + J_{ik} \bar{B}_{ik} \bar{A}_{jk}^*.$   $\tag{81}$

Hence, we get

$$[F_1 + F_2]_{ij} = \sum_{k=1}^D J_{jk} (G_{ik} - G_{ij}) \bar{A}_{ik} \bar{B}_{jk}^* \tag{82}$$

$$+ \sum_{k=1}^D J_{ik} (G_{jk} - G_{ij}) \bar{B}_{ik} \bar{A}_{jk}^*.$$

• Assume  $i \neq j$ . We have

$$[F_3]_{ij} = -[G_{ii} \bar{B}_{ii} - G_{jj} \bar{B}_{jj} - G_{ij} (\bar{B}_{ii} - \bar{B}_{jj})] J_{ij} \bar{A}_{ij} \tag{83}$$

$$= J_{ij} (G_{jj} - G_{ij}) \bar{B}_{jj} \bar{A}_{ij} - J_{ij} (G_{ii} - G_{ij}) \bar{A}_{ij} \bar{B}_{ii}. \tag{84}$$

For  $k \neq i$  and  $k \neq j$ , we have

$$J_{ik} (G_{jk} - G_{ij}) - J_{ij} (G_{jk} - G_{ik}) \tag{85}$$

$$= \frac{e^{\lambda_j} - e^{\lambda_k}}{(\lambda_j - \lambda_k)(\lambda_k - \lambda_i)} - \frac{e^{\lambda_i} - e^{\lambda_j}}{(\lambda_i - \lambda_j)(\lambda_k - \lambda_i)} \tag{86}$$

$$- \frac{e^{\lambda_j} - e^{\lambda_k}}{(\lambda_j - \lambda_k)(\lambda_j - \lambda_i)} + \frac{e^{\lambda_i} - e^{\lambda_k}}{(\lambda_i - \lambda_k)(\lambda_j - \lambda_i)}$$

$$= \frac{(\lambda_i - \lambda_j)(e^{\lambda_j} - e^{\lambda_k}) - (\lambda_j - \lambda_k)(e^{\lambda_i} - e^{\lambda_j})}{(\lambda_i - \lambda_j)(\lambda_j - \lambda_k)(\lambda_k - \lambda_i)} \tag{87}$$

$$+ \frac{(\lambda_k - \lambda_i)(e^{\lambda_j} - e^{\lambda_k}) + (\lambda_j - \lambda_k)(e^{\lambda_i} - e^{\lambda_k})}{(\lambda_i - \lambda_j)(\lambda_j - \lambda_k)(\lambda_k - \lambda_i)}$$

$$= \frac{e^{\lambda_i}(\lambda_k - \lambda_j + \lambda_j - \lambda_k)}{(\lambda_i - \lambda_j)(\lambda_j - \lambda_k)(\lambda_k - \lambda_i)} \tag{88}$$

$$+ \frac{e^{\lambda_j}(\lambda_i - \lambda_j + \lambda_j - \lambda_k + \lambda_k - \lambda_i)}{(\lambda_i - \lambda_j)(\lambda_j - \lambda_k)(\lambda_k - \lambda_i)}$$

$$+ \frac{e^{\lambda_k}(\lambda_j - \lambda_i + \lambda_i - \lambda_k + \lambda_k - \lambda_j)}{(\lambda_i - \lambda_j)(\lambda_j - \lambda_k)(\lambda_k - \lambda_i)} \tag{89}$$

$$= 0.$$

Similarly, we have  $J_{jk}(G_{ik} - G_{ij}) = J_{ij}(G_{jk} - G_{ik})$ . Hence, we get the following

$$[F_1 + F_2]_{ij} = \sum_{k \neq i, k \neq j} J_{ij} (G_{jk} - G_{ik}) (\bar{A}_{ik} \bar{B}_{jk}^* + \bar{B}_{ik} \bar{A}_{jk}^*) - J_{ij} (G_{ii} - G_{ij}) \bar{A}_{ii} \bar{B}_{ji}^* + J_{ij} (G_{jj} - G_{ij}) \bar{B}_{ij} \bar{A}_{jj}^*. \tag{90}$$

It follows that

$$[F_1 + F_2 + F_3]_{ij} \tag{91}$$

$$= \sum_{k \neq i, k \neq j} J_{ij} (G_{jk} - G_{ik}) (\bar{A}_{ik} \bar{B}_{jk}^* + \bar{B}_{ik} \bar{A}_{jk}^*)$$

$$+ J_{ij} (G_{ji} - G_{ii}) (\bar{A}_{ii} \bar{B}_{ji}^* + \bar{B}_{ii} \bar{A}_{ji}^*)$$

$$+ J_{ij} (G_{jj} - G_{ij}) (\bar{A}_{ij} \bar{B}_{jj}^* + \bar{B}_{ij} \bar{A}_{jj}^*)$$

$$= \sum_{k=1}^D \underbrace{J_{ij} (G_{jk} - G_{ik})}_{\phi_{i,j,k}} (\bar{A}_{ik} \bar{B}_{jk}^* + \bar{B}_{ik} \bar{A}_{jk}^*). \tag{92}$$

• Now assume that  $i = j$ . We have  $[F_3]_{ii} = G_{ii} \bar{A}_{ii} \bar{B}_{ii}$ . It follows that

$$[F_1 + F_2 + F_3]_{ii} \tag{93}$$



$$= \sum_{k=1}^D \mathbf{J}_{ik}(\mathbf{G}_{ik} - \mathbf{G}_{ii})(\bar{\mathbf{A}}_{ik}\bar{\mathbf{B}}_{ik}^* + \bar{\mathbf{B}}_{ik}\bar{\mathbf{A}}_{ik}^*) \quad (94)$$

$$+ \mathbf{G}_{ii}\bar{\mathbf{A}}_{ii}\bar{\mathbf{B}}_{ii} \quad (95)$$

$$= \sum_{k \neq i} \underbrace{\mathbf{J}_{ik}(\mathbf{G}_{ik} - \mathbf{G}_{ii})}_{\phi_{i,i,k}}(\bar{\mathbf{A}}_{ik}\bar{\mathbf{B}}_{ik}^* + \bar{\mathbf{B}}_{ik}\bar{\mathbf{A}}_{ik}^*) \quad (96)$$

$$+ \underbrace{\frac{1}{2}\mathbf{G}_{ii}(\bar{\mathbf{A}}_{ii}\bar{\mathbf{B}}_{ii}^* + \bar{\mathbf{B}}_{ii}\bar{\mathbf{A}}_{ii}^*)}_{\phi_{i,i,i}}$$

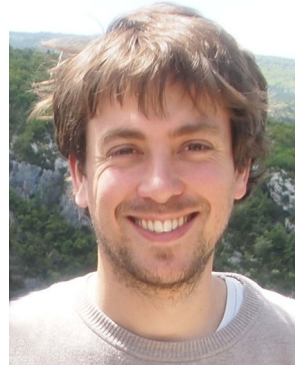
which concludes the proof.  $\square$

## References

- Aubert, G., Aujol, J.F.: A variational approach to removing multiplicative noise. *SIAM J. Appl. Math.* **68**(4), 925–946 (2008)
- Baqué, R., du Plessis, O.R., Castet, N., Fromage, P., Martinot-Lagarde, J., Nouvel, J.F., Oriot, H., Angelliaume, S., Brigui, F., Cantalloube, H., et al.: SETHI/RAMSES-NG: New performances of the flexible multi-spectral airborne remote sensing research platform. In: 2017 European Radar Conference (EURAD), pp. 191–194. IEEE (2017)
- Boyd, S., Parikh, N., Chu, E.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Now Publishers Inc (2011)
- Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.* **4**(2), 490–530 (2005)
- Candes, E.J., Sing-Long, C.A., Trzasko, J.D.: Unbiased risk estimates for singular value thresholding and spectral estimators. *IEEE Trans. Signal Process.* **61**(19), 4643–4657 (2013)
- Chan, S.H., Wang, X., Elgandy, O.A.: Plug-and-play ADMM for image restoration: fixed-point convergence and applications. *IEEE Trans. Comput. Imag.* **3**(1), 84–98 (2016)
- Chen, J., Chen, Y., An, W., Cui, Y., Yang, J.: Nonlocal filtering for polarimetric SAR data: A pretest approach. *IEEE Trans. Geosci. Remote Sens.* **49**(5), 1744–1754 (2010)
- Chierchia, G., Cozzolino, D., Poggi, G., Verdoliva, L.: SAR image despeckling through convolutional neural networks. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 5438–5441. IEEE (2017)
- Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing. In: Fixed-point algorithms for inverse problems in science and engineering, pp. 185–212. Springer (2011)
- Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
- Dalsasso, E., Denis, L., Tupin, F.: As if by magic: self-supervised training of deep despeckling networks with merlin. *IEEE Trans. Geosci. Remote Sens.* (2021). <https://doi.org/10.1109/TGRS.2021.3128621>
- Dalsasso, E., Denis, L., Tupin, F.: Sar2sar: a semi-supervised despeckling algorithm for sar images. *IEEE J. Select. Top. Appl. Earth Observat. Remote Sens.* (2021)
- Dalsasso, E., Yang, X., Denis, L., Tupin, F., Yang, W.: SAR image despeckling by deep neural networks: from a pre-trained model to an end-to-end training strategy. *Remote Sens.* **12**(16), 2636 (2020)
- Deledalle, C.A., Denis, L., Tabti, S., Tupin, F.: MuLoG, or How to apply Gaussian denoisers to multi-channel SAR speckle reduction? *IEEE Trans. Image Process.* **26**(9), 4389–4403 (2017)
- Deledalle, C.A., Denis, L., Tupin, F.: Iterative weighted maximum likelihood denoising with probabilistic patch-based weights. *IEEE Trans. Image Process.* **18**(12), 2661–2672 (2009)
- Deledalle, C.A., Denis, L., Tupin, F.: NL-InSAR: nonlocal interferogram estimation. *IEEE Trans. Geosci. Remote Sens.* **49**(4), 1441–1452 (2010)
- Deledalle, C.A., Denis, L., Tupin, F., Reigber, A., Jäger, M.: NL-SAR: a unified nonlocal framework for resolution-preserving (Pol)(In) SAR denoising. *IEEE Trans. Geosci. Remote Sens.* **53**(4), 2021–2038 (2014)
- Deledalle, C.A., Vaiter, S., Peyré, G., Fadili, J.M., Dossal, C.: Risk estimation for matrix recovery with spectral regularization. In: ICML’2012 workshop on Sparsity, Dictionaries and Projections in Machine Learning and Signal Processing (2012)
- Denis, L., Tupin, F., Darbon, J., Sigelle, M.: SAR image regularization with fast approximate discrete minimization. *IEEE Trans. Image Process.* **18**(7), 1588–1600 (2009)
- Durand, S., Fadili, J., Nikolova, M.: Multiplicative noise removal using L1 fidelity on frame coefficients. *J. Math. Imag. Vis.* **36**(3), 201–226 (2010)
- Even, M., Schulz, K.: InSAR deformation analysis with distributed scatterers: A review complemented by new advances. *Remote Sensing* **10**(5), 744 (2018)
- Goodman, J.: Some fundamental properties of speckle. *J. Opt. Soc. Am.* **66**(11), 1145–1150 (1976)
- Goodman, J.W.: Statistical properties of laser speckle patterns. In: Laser speckle and related phenomena, pp. 9–75. Springer (1975)
- Guo, B., Han, Y., Wen, J.: Agem: Solving linear inverse problems via deep priors and sampling. *Adv. Neural. Inf. Process. Syst.* **32**, 547–558 (2019)
- Hertrich, J., Neumayer, S., Steidl, G.: Convolutional proximal neural networks and plug-and-play algorithms. *Linear Algebra and its Applications* (2021)
- Kadkhodaie, Z., Simoncelli, E.P.: Solving linear inverse problems using the prior implicit in a denoiser. *arXiv preprint arXiv:2007.13640* (2020)
- Kawar, B., Vaksman, G., Elad, M.: Stochastic image denoising by sampling from the posterior distribution. *arXiv preprint arXiv:2101.09552* (2021)
- Knaus, C., Zwicker, M.: Dual-domain image denoising. In: 2013 IEEE International Conference on Image Processing, pp. 440–444. IEEE (2013)
- Laumont, R., De Bortoli, V., Almansa, A., Delon, J., Durmus, A., Pereyra, M.: Bayesian imaging using Plug & Play priors: when Langevin meets Tweedie. *arXiv preprint arXiv:2103.04715* (2021)
- Lee, J.S.: Digital image smoothing and the sigma filter. *Comput. Vis. Graph. Image Process.* **24**(2), 255–269 (1983)
- Lewis, A.S., Sendov, H.S.: Twice differentiable spectral functions. *SIAM J. Matrix Anal. Appl.* **23**(2), 368–386 (2001)
- Lopes, A., Touzi, R., Nezry, E.: Adaptive speckle filters and scene heterogeneity. *IEEE Trans. Geosci. Remote Sens.* **28**(6), 992–1000 (1990)
- Meinhardt, T., Moller, M., Hazirbas, C., Cremers, D.: Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1781–1790 (2017)
- Molini, A.B., Valsesia, D., Fracastoro, G., Magli, E.: Speckle2Void: deep self-supervised SAR despeckling with blind-spot convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* (2021)
- Monga, V., Li, Y., Eldar, Y.C.: Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Process. Mag.* **38**(2), 18–44 (2021)
- Moreira, A., Prats-Iraola, P., Yoonis, M., Krieger, G., Hajnsek, I., Papathanassiou, K.P.: A tutorial on synthetic aperture radar. *IEEE Geosci. Remote Sens. Magaz.* **1**(1), 6–43 (2013)

37. Parameswaran, S., Deledalle, C.A., Denis, L., Nguyen, T.Q.: Accelerating GMM-based patch priors for image restoration: three ingredients for a 100× speed-up. *IEEE Trans. Image Process.* **28**(2), 687–698 (2018)
38. Parrilli, S., Poderico, M., Angelino, C.V., Verdoliva, L.: A nonlocal SAR image denoising algorithm based on LLMMSE wavelet shrinkage. *IEEE Trans. Geosci. Remote Sens.* **50**(2), 606–616 (2011)
39. Rasti, B., Chang, Y., Dalsasso, E., Denis, L., Ghamisi, P.: Image restoration for remote sensing: overview and toolbox. *IEEE Geosci. Remote Sens. Magaz.* 2–31 (2021). <https://doi.org/10.1109/MGRS.2021.3121761>
40. Reehorst, E.T., Schniter, P.: Regularization by denoising: clarifications and new interpretations. *IEEE Trans. Comput. Imaging* **5**(1), 52–67 (2018)
41. Romano, Y., Elad, M., Milanfar, P.: The little engine that could: regularization by denoising (red). *SIAM J. Imag. Sci.* **10**(4), 1804–1844 (2017)
42. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**(1–4), 259–268 (1992)
43. Ryu, E., Liu, J., Wang, S., Chen, X., Wang, Z., Yin, W.: Plug-and-play methods provably converge with properly trained denoisers. In: *International Conference on Machine Learning*, pp. 5546–5557. PMLR (2019)
44. Steidl, G., Teuber, T.: Removing multiplicative noise by Douglas-Rachford splitting methods. *J. Math. Imag. Vis.* **36**(2), 168–184 (2010)
45. Terris, M., Repetti, A., Pesquet, J.C., Wiaux, Y.: Enhanced convergent pnp algorithms for image restoration. In: *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 1684–1688. IEEE (2021)
46. Touzi, R., Lopes, A., Bruniquel, J., Vachon, P.W.: Coherence estimation for SAR imagery. *IEEE Trans. Geosci. Remote Sens.* **37**(1), 135–149 (1999)
47. Vasile, G., Trouvé, E., Lee, J.S., Buzuloiu, V.: Intensity-driven adaptive-neighborhood technique for polarimetric and interferometric SAR parameters estimation. *IEEE Trans. Geosci. Remote Sens.* **44**(6), 1609–1621 (2006)
48. Venkatakrisnan, S.V., Bouman, C.A., Wohlberg, B.: Plug-and-play priors for model based reconstruction. In: *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948. IEEE (2013)
49. Xie, H., Pierce, L.E., Ulaby, F.T.: Statistical properties of logarithmically transformed speckle. *IEEE Trans. Geosci. Remote Sens.* **40**(3), 721–727 (2002)
50. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.* **26**(7), 3142–3155 (2017)
51. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep CNN denoiser prior for image restoration. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 3929–3938 (2017)
52. Zhang, Y., Zhu, D.: Height retrieval in postprocessing-based VideoSAR image sequence using shadow information. *IEEE Sens. J.* **18**(19), 8108–8116 (2018)
53. Zhao, W., Deledalle, C.A., Denis, L., Maître, H., Nicolas, J.M., Tupin, F.: Ratio-based multitemporal SAR images denoising: RABASAR. *IEEE Trans. Geosci. Remote Sens.* **57**(6), 3552–3565 (2019)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Charles-Alban Deledalle** received the Engineering degree from EPITA and the Master of Science degree from Univ. Paris VI both in France, in 2008. In 2011, he defended his PhD from LTCI, Telecom Paris-Tech, France, in Signal and Image Processing. He made a postdoctoral fellowship in Applied Mathematics at CEREMADE, Univ. Paris IX, France, in 2011–2012. He was a CNRS Associate Scientist at IMB, Univ. Bordeaux, France, in 2012–2019, during which he did a sabbatical as a visiting professor at UC San Diego in the ECE department from 2017 to 2019. He has been a Research Scientist/Engineer in Machine Learning in the Research and Development Department at Brain Corporation in San Diego since 2020. He was the recipient of the IEEE ICIP Best Student Paper Award in 2010, the ISIS/EEA/GRETSI Best PhD Award in 2012, the IEEE GRSS Transactions Prize Paper Award in 2016, and the UCSD ECE Best Lecturer Award in 2019.



**Loïc Denis** is Associate Professor at the University Jean Monnet, Saint-Etienne, France. His research interests include image denoising and reconstruction, robust signal processing, source detection, and machine learning with applications in remote sensing, diffractive microscopy and astronomy. He is a co-recipient of IEEE ICIP best student paper award (2010), EURASIP EUSIPCO best student paper award (2015) and the IEEE Geoscience and Remote Sensing Society Transactions Prize Paper Award (2016).



**Florence Tupin** (Senior Member, IEEE) received the engineering degree and the Ph.D. degree in signal and image processing from Ecole Nationale Supérieure des Telecommunications (ENST), Paris, France, in 1994 and 1997, respectively, and the Habilitation à Diriger des Recherches degree from the University of Rennes, Rennes, France, in 2007. From 1997 to 1998, she was with SAGEM, Paris, France, where she worked on fingerprint recognition. She is currently a Professor of image and signal processing with LTCI, Telecom Paris, Institut Polytechnique de Paris, France. Since 2014, she has been the Head of the Image, Modeling, Analysis, GEometry, and Synthesis Team of LTCI. She has coauthored more than 200 papers. Her research interests include image processing and analysis, especially for remote sensing and synthetic aperture radar imaging applications, and earth observation. Dr. Tupin has been a member of several international and national technical conference committees since 2003. She was the Chair of the Urban Remote Sensing Joint Event held in Paris in

2007. She was an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING from 2007 to 2016. She was the recipient of several awards, among them the IEEE GRSS

Transactions Prize Paper Award in 2016 for work on nonlocal speckle filtering.