IPOL IMAGE PROCESSING ON LINE

Published in Image Processing On Line on 2022–04–25.
Submitted on 2021–04–16, accepted on 2022–04–11.
ISSN 2105–1232 © 2022 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2022.346

Non-Local Means Filters for Full Polarimetric Synthetic Aperture Radar Images with Stochastic Distances

Luis Gomez[1], Jie Wu[2], Alejandro C. Frery[3]

[1] Department of Electronic Engineering and Automatic Control, Universidad de Las Palmas de Gran Canaria, Spain (luis.gomez@ulpgc.es)
[2] School of Computer Science, Shaanxi Normal University, China (wujie_1985@snnu.edu.cn)
[3] School of Mathematics and Statistics, Victoria University of Wellington, New Zealand (alejandro.frery@vuw.ac.nz)

Communicated by Thibaud Ehret *Demo edited by* Thibaud Ehret

Abstract

In this paper, we present a Non-Local Means filtering method for PolSAR (Polarimetric Synthetic Aperture Radar) imagery. We adopted three stochastic distances to measure the similarity of samples. We obtain p-values using the asymptotic distribution of test statistics based on these distances. Then we use this similarity evidence as input for a smooth activating function that yields the weights of local convolution matrices. Non-local filters are computationally demanding, so we provide an efficient implementation that allows us to experiment with different settings and find optimal parameters.

Source Code

The source code (ANSI C++), its documentation, and the online demo are accessible at the IPOL the web page of this article[1]. Compilation and usage instructions are included in the `README.txt` file of the archive.

Keywords: stochastic distances; non-local means; Wishart distribution; PolSAR imagery

1 Introduction

The local mean is the simplest noise reduction filter in image processing. It amounts to applying the same convolution to every pixel with a typically small, e.g., 3×3, 5×5, 7×7 pixels, mask of equal and positive weights. Such a filter effectively reduces the noise, it is easily implemented, but it also introduces blurring. One may mention the Gaussian smoothing among the alternatives that retain the simplicity and the ability to reduce the effect of noise. The weights are proportional to a normal density with zero mean and pre-fixed variance in this filter.

[1] https://doi.org/10.5201/ipol.2022.346

Those two filters belong to the class of translation-invariant convolutional filters, for which there is a wealth of knowledge. See, for instance, the book by Lim [19]. Although it is possible to design filters with good properties, their translation-invariant nature prevents them from adapting to different situations.

Buades et al. [2, 3] proposed an adaptive convolutional approach: the Non-Local Means (NLM), in which "Non-Local" refers to the use of large convolutional windows. This class of filters has produced remarkable results in the denoising of natural [15, 20], Synthetic Aperture Radar – SAR [6, 7, 23, 31], and medical images [28], among other applications.

While the design of a translation-invariant convolutional filter mainly relies on the analysis of its Fourier spectrum, NLM filters are defined by two simple concepts: (i) how can we capture the similarity between two samples, and (ii) how can we transform the similarity into a mask's weight.

In the original proposal, by extending a single pixel to a local patch, Buades et al. [2] used a Gaussian weighted Euclidean distance between image patches to measure their similarity. Such a measure has good properties with additive white noise but is questionable in the presence of other types of contamination. Besides, the mapping between similarities and weights is implicit, leaving little control to the designer.

Speckle noise, although a deterministic interference pattern, can be adequately modeled as a stochastic process [10]. Such noise is neither additive nor Gaussian and is common to all images obtained with coherent illumination. Gao [12] and Yue et al. [35] provide a survey of distributions for univariate data, while Deng et al. [8] discuss, from a physical perspective, distributions for polarimetric observations.

Nascimento et al. [22] obtained tests statistics for the null hypothesis that the same distribution produced two samples under a model for SAR data: the Multiplicative Model. Such test statistics stem from computing stochastic divergences between the samples and, thus, consider the peculiarities of these data. This first work handled univariate (intensity) data, and later Frery et al. [11] extended the approach to the complex-valued multivariate case of polarimetric images. In both cases, the approach was using the family of h-ϕ divergences to obtain test statistics [24, 25, 27]. Such information-theoretic tools allow obtaining an arbitrary number of test statistics with the same asymptotic distribution. We review these divergences and tests in Section 2.3.

Torres et al. [30], Deledalle et al. [6], Wu et al. [33] and Vitale et al. [32] discuss ways of computing similarities in the presence of speckle. Chan et al. [4] show that the Shannon entropy can also be used in the comparison. These works designed different similarities to compute the weights of NLM filters.

Gomez-Deniz and Frery [13] analyzed the effectiveness of three different stochastic distances (namely, Hellinger, Kullback-Leibler, and Bhattacharyya) for speckle reduction in PolSAR imagery using NLM filters. This paper pursues this line of research by providing an interactive tool that allows experimenting with different setups (kinds of distances and tuning parameters) for PolSAR speckle reduction. Our implementation alleviates the computational burden that NLM filters impose.

The rest of this paper is organized as follows. Section 2 presents an overview of NLM filters and stochastic distances under the complex Wishart distribution for PolSAR data. Section 3 provides details of our implementation. In Section 4, we present the set of metrics used to evaluate the quality of the results. Section 5 contains results obtained with different parameter settings and both simulated PolSAR data and actual images. Section 6 presents the main conclusions. Specific details of the codes are provided in the Appendix.

2 Definitions and Notation

2.1 The Non-Local Means Approach to Noise Reduction

Figure 1 illustrates the idea of statistical non-local means in which we want to estimate $9 \times 9 - 1 = 80$ weights for the convolution mask (right grid) that will act around the central pixel (red double dash, left grid). The weight w_1 is typically a non-decreasing function of p_1, the p-value of the test statistic that assesses the null hypothesis that the samples $\boldsymbol{z}^{(0)} = (\boldsymbol{z}_1^{(0)}, \boldsymbol{z}_2^{(0)}, \ldots, \boldsymbol{z}_9^{(0)})$ (red dashes) and $\boldsymbol{z}^{(1)} = (\boldsymbol{z}_1^{(1)}, \boldsymbol{z}_2^{(1)}, \ldots, \boldsymbol{z}_{25}^{(1)})$ (orange dashes) were produced by the same probability law $\mathcal{D}(\theta)$, where θ is the parameter that indexes the distribution. This statistical formulation allows samples of different sizes, i.e. different estimation windows, cf. the orange and green patches.

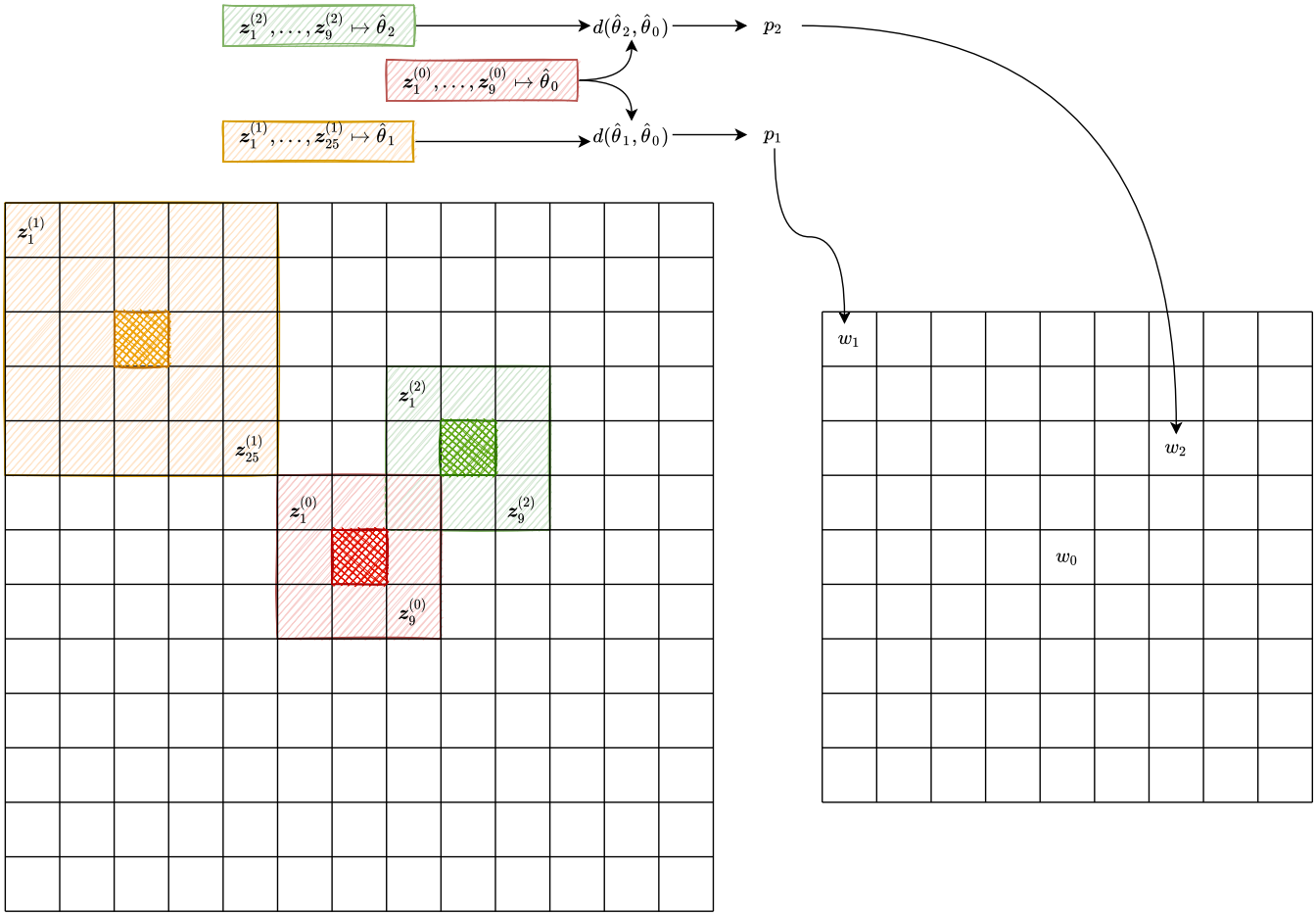


Figure 1: Sketch of the statistical non-local means approach.

The sample $\boldsymbol{z}^{(0)}$ which surrounds and includes the central pixel, is transformed into the estimate $\widehat{\theta}_0$. The sample $\boldsymbol{z}^{(1)}$ which surrounds and includes the pixel corresponding to w_1 (orange double dash) is transformed into the estimate $\widehat{\theta}_1$. These estimates are compared by a measure of dissimilarity between the models $d(\widehat{\theta}_0, \widehat{\theta}_1)$. Such a measure, in the context of h-ϕ divergences, is turned into a test statistic for the null hypothesis

> H_0: the random vectors $\boldsymbol{Z}_1, \boldsymbol{Z}_2$ that gave rise to the samples $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}$ obey the same distribution $\mathcal{D}(\theta)$.

Knowing the (possibly asymptotic) distribution of such test statistic, we compute p_1, the p-value under H_0, and then we transform it into the desired weight w_1.

Analogously, the weight w_2 is the result of verifying the hypothesis that the random vectors $\boldsymbol{Z}_0, \boldsymbol{Z}_2$ that produced the samples $\boldsymbol{z}^{(0)}$ and $\boldsymbol{z}^{(2)} = (\boldsymbol{z}_1^{(2)}, \boldsymbol{z}_2^{(2)}, \ldots, \boldsymbol{z}_9^{(2)})$ (green dashes) follow the same distribution. Notice that the estimates $\widehat{\theta}$ are computed only once for every pixel.

Each p-value may be used directly as a weight w but, as discussed by Torres et al. [31], such a choice introduces a conceptual distortion. Consider, for instance, the samples $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$. When contrasted with the central sample $\boldsymbol{z}^{(0)}$ they produced the p-values $p_1 = 0.05$ and $p_2 = 0.93$. In this case, the first weight is significantly smaller than the second one, but there is no evidence to reject the hypothesis at the confidence level $\eta = 0.05$.

Torres et al. [31] proposed using a piece-wise linear function that maps all values above η to 1, a linear transformation of values between $\eta/2$ and η, and zero below $\eta/2$. This work proposes a smooth activating function with zero first- and second-order derivatives at the inflection points. The "smoother step function" defined in [9] is given as

$$f_{\text{smoother}}(x) = \begin{cases} 0 & \text{if } x < 0, \\ 6x^5 - 15x^4 + 10x^3 & \text{if } 0 \leq x \leq 1, \\ 1 & \text{if } x > 1. \end{cases}$$

This function connects in a smooth manner the points $(0, 0)$ and $(1, 1)$. We use f_{smoother} to connect $(\eta/k, 0)$ and $(\eta, 1)$, for every $k > 1$, and define the weight w as a function of the observed p-value as

$$w(p) = f_{\text{smoother}}\left(\frac{p - \eta/k}{\eta - \eta/k}\right). \tag{1}$$

This function is zero for $p < \eta/k$, and is one above η. The parameter k controls the steepness of the transformation, as shown in Figure 2. From our experiments, we recommend $k = 2$.

Figure 2 illustrates $w(p)$ for $\eta = 0.7$ and $k = 2, 3, 10$.

2.2 PolSAR Imaging and the Wishart Distribution

PolSAR data result from the interaction between the electromagnetic signal emitted by the sensor and the target. The sensor then captures the return as a complex-valued matrix $\boldsymbol{S}'$

$$\boldsymbol{S}' = \begin{bmatrix} S_{\text{HH}} & S_{\text{HV}} \\ S_{\text{VH}} & S_{\text{VV}} \end{bmatrix},$$

in which the indexes "H" and "V" denote Horizontal and Vertical polarization. Under mild conditions, it is possible to assume that the reciprocity theorem holds [18] and, thus, $S_{\text{HV}} = S_{\text{VH}}$. All the necessary information is, therefore, contained in the (complex) scattering vector

$$\boldsymbol{S} = \begin{bmatrix} S_{\text{HH}} & S_{\text{HV}} & S_{\text{HH}} \end{bmatrix}^{\text{T}}, \tag{2}$$

where "T" denotes transposition.

The joint distribution of $\boldsymbol{S}$ depends on, among other factors, the properties, number and spatial distribution of the elementary backscatters in each resolution cell [34, 35].

Goodman [14] proved that when the area under observation is comprised of a massive number of scatterers, and when none dominates the return, then (2) may be described by a zero-mean complex Gaussian distribution. Different types of targets reflect on the entries of the (complex-valued) covariance matrix $\boldsymbol{\Sigma}$ that indexes this distribution.

Dealing directly with scattering vectors data is not as usual as working with multilooked data. This is because Multilook processing enhances the signal-to-noise ratio. This operation consists

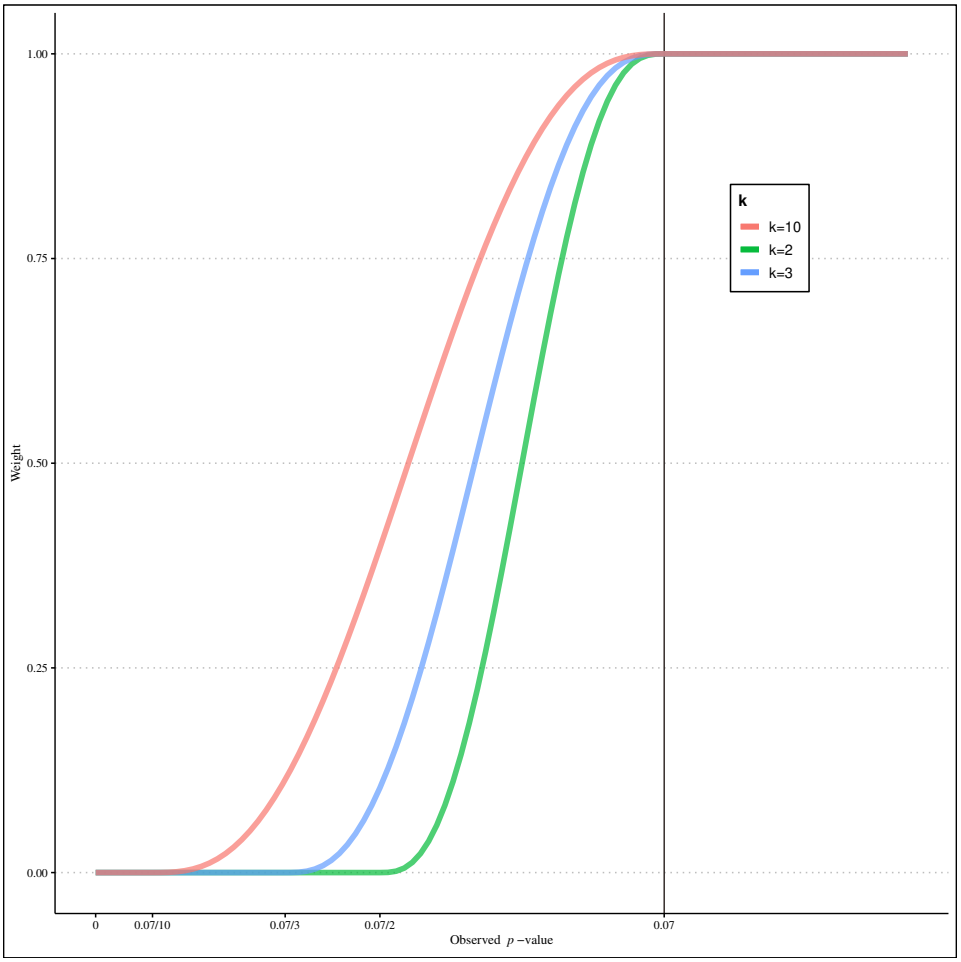


Figure 2: The transformation between p-values and weights for $\eta = 0.07$ and $k \in \{2, 3, 10\}$.

in forming the average of (complex-valued) matrices with L ideally independent and identically distributed scattering vectors

$$\boldsymbol{Z} = \frac{1}{L}\sum_{\ell=1}^{L} \boldsymbol{S}(\ell)\boldsymbol{S}^{\mathrm{T}*}(\ell) = \frac{1}{L}\sum_{\ell=1}^{L} \begin{bmatrix} S_{\mathrm{HH}}(\ell)S^*_{\mathrm{HH}}(\ell) & S_{\mathrm{HH}}(\ell)S^*_{\mathrm{HV}}(\ell) & S_{\mathrm{HH}}(\ell)S^*_{\mathrm{VV}}(\ell) \\ S_{\mathrm{HV}}(\ell)S^*_{\mathrm{HH}}(\ell) & S_{\mathrm{HV}}(\ell)S^*_{\mathrm{HV}}(\ell) & S_{\mathrm{HV}}(\ell)S^*_{\mathrm{VV}}(\ell) \\ S_{\mathrm{VV}}(\ell)S^*_{\mathrm{HH}}(\ell) & S_{\mathrm{VV}}(\ell)S^*_{\mathrm{HV}}(\ell) & S_{\mathrm{VV}}(\ell)S^*_{\mathrm{VV}}(\ell) \end{bmatrix} = \\ = \begin{bmatrix} I_{\mathrm{HH}} & \mathrm{Cov}_{\mathrm{HH,HV}} & \mathrm{Cov}_{\mathrm{HH,VV}} \\ \mathrm{Cov}_{\mathrm{HH,HV}} & I_{\mathrm{HV}} & \mathrm{Cov}_{\mathrm{HH,VV}} \\ \mathrm{Cov}_{\mathrm{HH,VV}} & \mathrm{Cov}_{\mathrm{HH,VV}} & I_{\mathrm{VV}} \end{bmatrix}, \quad (3)$$

where "$*$" denotes the complex conjugate. The (real-valued) diagonal elements of (3) are referred to as "Intensity Channels", and the off-diagonal elements are the "Complex Covariance Channels".

Two things are noteworthy. First, despite the usual denomination of (3) as "covariance matrix," it is a random variable. Second, the expected value of $\boldsymbol{Z}$ is $\mathrm{E}(\boldsymbol{Z}) = \boldsymbol{\Sigma}$.

Goodman [14] also obtained the distribution of $\boldsymbol{Z}$ under the assumption that $\boldsymbol{S}(1), \boldsymbol{S}(2), \ldots, \boldsymbol{S}(L)$ are independent identically distributed zero-mean complex Gaussian random variables. The distribution of $\boldsymbol{Z}$ is characterized by the following probability density function

$$f_{\boldsymbol{Z}}(\boldsymbol{z}; \boldsymbol{\Sigma}, L) = \frac{L^{3L}|\boldsymbol{z}|^{L-3}}{|\boldsymbol{\Sigma}|^{L}\Gamma_3(L)} \exp\left\{-L\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{z})\right\}, \quad (4)$$

where $L \geq 3$, $\Gamma_3(L) = \pi^3 \prod_{i=0}^{2} \Gamma(L-i)$, $\mathrm{Tr}(\cdot)$ is the trace operator, and $\boldsymbol{z}$ spans the whole set of positive-definite Hermitian 3×3 matrices $\mathcal{A}$. This is the complex scaled multilook Wishart

distribution, which we denote $\boldsymbol{Z} \sim \mathcal{W}(\boldsymbol{\Sigma}, L)$ and that we will call "Wishart law". Lee et al. [17] and Hagedorn et al. [16] have studied some of the marginal distributions that arise from this model.

Given the random sample $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n$ of independent identically distributed $\mathcal{W}(\boldsymbol{\Sigma}, L)$ deviates, the maximum likelihood estimator $(\widehat{\boldsymbol{\Sigma}}, \widehat{L})$ is given by

$$\text{the sample mean: } \widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i, \tag{5}$$

$$\text{and the root of: } 3\ln \widehat{L} + \frac{1}{n}\sum_{i=1}^{n} \ln|\boldsymbol{z}_i| - \ln|\widehat{\boldsymbol{\Sigma}}| - \psi_3^{(0)}(\widehat{L}) = 0, \tag{6}$$

where $\psi_3^{(0)}$ is the zero-order term of the ν-th order multivariate polygamma function $\psi_3^{(0)}(L) = \sum_{i=0}^{2} \psi^{(0)}(L-1)$, and $\psi^{(0)}$ is the digamma function

$$\psi^{(0)}(L) = \frac{\partial \ln \Gamma(L)}{\partial L}.$$

To numerically solve (6), $\widehat{L}$ is initialized to the nominal number of looks of the PolSAR data, which is a known value. The estimated value, $\widehat{L}$ is also known as ENL or *Equivalent Number of Looks*.

2.3 Stochastic Divergences, Distances and Tests

Posed in terms of the non-local means filter situation, Frery [10] noted that this and several other relevant problems in signal and image processing and analysis could be solved by formulating them in the form of the following hypothesis test:

> Consider the samples $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$ of the form $\boldsymbol{z}^{(j)} = \boldsymbol{z}_1^{(j)}, \boldsymbol{z}_2^{(j)}, \ldots, \boldsymbol{z}_{n_j}^{(j)}$ for $j = 1, 2$. Is there enough evidence to reject the null hypothesis that the same model $\mathcal{D}(\theta)$ produced them?

This problem assumes that the samples might be of different sizes n_1, n_2, and that the model $\mathcal{D}(\theta)$ is a probability distribution indexed by the unknown q-dimensional parameter $\theta \in \Theta \subset \mathbb{R}^q$.

On the one hand, we deal with complex-valued observations in matrix form, so usual techniques are seldom helpful. On the other hand, we have an adequate distribution (i.e. the Wishart law) to model these data. Thus, we will rely on the approach based on stochastic divergences between Wishart models.

Let X and Y be random variables defined over the same probability space and same support $\mathcal{A}$. Assume their distributions are characterized by the densities $f_X(v;\theta_1)$ and $f_Y(v;\theta_2)$, respectively, where θ_1 and θ_2 are parameter vectors. Consider $\phi\colon (0,\infty) \to [0,\infty)$ a convex function, $h\colon (0,\infty) \to [0,\infty)$ a strictly increasing function with $h(0) = 0$, and indeterminate forms are assigned value zero. The (h,ϕ)-divergence between f_X and f_Y is defined by

$$D_\phi^h(X,Y) = h\left(\int_{\mathcal{A}} \phi\left(\frac{f_X(v;\theta_1)}{f_Y(v;\theta_2)}\right) f_Y(v;\theta_2)\,\mathrm{d}v\right). \tag{7}$$

Table 1 presents some well-known divergence measures, constructed by choosing adequately h and ϕ.

Some divergence measures lack symmetry. Although there are numerous methods to address the symmetry problem [29], a simple solution is to define a new measure d_ϕ^h given by

$$d_\phi^h(X,Y) = \frac{D_\phi^h(X,Y) + D_\phi^h(Y,X)}{2}, \tag{8}$$

(h,ϕ)-divergence	$h(y)$	$\phi(x)$
Kullback-Leibler	y	$x\ln(x)$
Hellinger	$y/2, 0 \leq y < 2$	$(\sqrt{x}-1)^2$
Bhattacharyya	$-\ln(1-y), 0 \leq y < 1$	$-\sqrt{x}+\frac{x+1}{2}$

Table 1: (h,ϕ)-divergences and related functions ϕ and h, from [10, 11]

regardless whether $D_\phi^h(\cdot,\cdot)$ is symmetric or not. Henceforth, the symmetrical versions of the divergence measures are termed as "distances". We obtain integral formulas for the distance measures by applying the functions of Table 1 into Equation (7), and symmetrizing the resulting divergences with (8). For simplicity, in the list below, we suppress the explicit dependence on z and on the support $\mathcal{A}$.

- The Kullback-Leibler distance:

$$d_{\mathrm{KL}}(X,Y) = \frac{1}{2}\int (f_X - f_Y)\ln\frac{f_X}{f_Y}.$$

- The Hellinger distance:

$$d_{\mathrm{H}}(X,Y) = 1 - \int \sqrt{f_X f_Y}.$$

- The Bhattacharyya distance:

$$d_{\mathrm{B}}(X,Y) = -\ln\int\sqrt{f_X f_Y} = -\ln\left(1 - d_{\mathrm{H}}(X,Y)\right). \tag{9}$$

The distances mentioned above are neither comparable nor semantically rich. References [24, 25, 26] are pioneering works that make a connection between any h-ϕ distance and a test statistic.

Consider the model $\mathcal{D}(\theta)$, with $\theta \in \Theta \subset \mathbb{R}^q$ the unknown parameter that indexes the distribution. Assume the availability of two samples of i.i.d. observations $\boldsymbol{z}^{(1)} = \boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{n_1}$ from $\mathcal{D}(\theta_1)$, and $\boldsymbol{z}^{(2)} = \boldsymbol{z}_{n_1+1}, \boldsymbol{z}_{n_1+2}, \ldots, \boldsymbol{z}_{n_1+n_2}$ from $\mathcal{D}(\theta_2)$, $\theta_1, \theta_2 \in \Theta$. The maximum likelihood estimates of θ_1 and θ_2 are computed as $\widehat{\theta}_1(\boldsymbol{z}^{(1)})$ and $\widehat{\theta}_2(\boldsymbol{z}^{(2)})$. We are interested in verifying if there is enough evidence in $\widehat{\theta}_1, \widehat{\theta}_2$ to reject the null hypothesis $H_0 : \theta_1 = \theta_2$.

Under the regularity conditions discussed in Reference [27, p. 380] the following lemma holds:

Lemma 1. *If $n_1/(n_1+n_2)$ converges to a constant in $(0,1)$ when $n_1, n_2 \to \infty$, and $\theta_1 = \theta_2$, then*

$$S_\phi^h(\widehat{\theta}_1, \widehat{\theta}_2) = \frac{2n_1n_2}{n_1+n_2}\frac{d_\phi^h(\widehat{\theta}_1,\widehat{\theta}_2)}{h'(0)\phi''(1)} \xrightarrow{\mathcal{D}} \chi_q^2, \tag{10}$$

where "$\xrightarrow{\mathcal{D}}$" denotes convergence in distribution and χ_q^2 represents the chi-square distribution with q degrees of freedom.

Note that q is the dimension of θ. Based on Lemma 1, we obtain a test for the null hypothesis $\theta_1 = \theta_2$ in the form of the following proposition.

Proposition 2. *Let n_1 and n_2 be large and $S_\phi^h(\widehat{\theta}_1, \widehat{\theta}_2) = s$, then the null hypothesis $\theta_1 = \theta_2$ can be rejected at level α if $\Pr(\chi_q^2 > s) \leq \alpha$.*

This result holds for every h-ϕ divergence, hence its generality. The (asymptotic) p-value of the S_ϕ^h statistic can be used as a measure of the evidence against rejecting H_0, so it has rich semantic information. The p-value could be used to compute the weight of the non-local means filters. Moreover, since the limit distribution χ^2_q is the same for all h and ϕ, the test statistics S_ϕ^h are comparable.

Frery et al. [11] obtained explicit expressions for several stochastic distances between Wishart models. In particular, we cite the Kullback-Leibler and Bhattacharyya distances

$$d_{\mathrm{KL}}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2) = \frac{L_1-L_2}{2}\left\{\log\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|} - 3\log\frac{L_1}{L_2} + p\left[\psi^{(0)}(L_1-2)-\psi^{(0)}(L_2-2)\right] + (L_2-L_1)\sum_{i=1}^{2}\frac{i}{(L_1-i)(L_2-i)}\right\} + \frac{\mathrm{Tr}(L_2\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1 + L_1\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2)}{2} - \frac{3(L_1+L_2)}{2}, \tag{11}$$

$$d_{\mathrm{B}}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2) = \frac{L_1\log|\boldsymbol{\Sigma}_1|}{2} + \frac{L_2\log|\boldsymbol{\Sigma}_2|}{2} - \frac{L_1+L_2}{2}\log\left|\left(\frac{L_1\boldsymbol{\Sigma}_1^{-1}+L_2\boldsymbol{\Sigma}_2^{-1}}{2}\right)^{-1}\right| + \sum_{k=0}^{2}\log\frac{\sqrt{\Gamma(L_1-k)\Gamma(L_2-k)}}{\Gamma(\frac{L_1+L_2}{2}-k)} - \frac{3}{2}(L_1\log L_1 + L_2\log L_2). \tag{12}$$

The Hellinger distance d_{H} follows from (9) and (12). These three distances reduce to simple expressions under the usual condition $L_1 = L_2$ [11]. In such a case, $L_1 = L_2 = L$ (the nominal number of looks). However, it is recommended to estimate L_1 and L_2 by solving (6). Also, notice that these three distances depend only on the number of looks and on simple operations on the covariance matrix, namely the inverse and the determinant. Coelho et al. [5] proposed a fast algorithm for simultaneous inversion and determinant computation of these matrices.

Figure 3 shows the steps that transform two samples into a p-value under the Wishart model and the p-value into the weight of the convolution mask.

Notice that the Non-Local Means approach illustrated in Figure 1 violates the large sample hypothesis required by (10). The estimation windows (patches) are typically small to avoid blurring. Previous works, e.g. [11], have shown, though, that h-ϕ test statistics under the Wishart model are comparable to their asymptotic distribution even with very small samples. With those results in mind, the user may choose estimation windows of typical size, e.g. 3×3, 5×5, etc., and still count on the validity of the observed p-values.

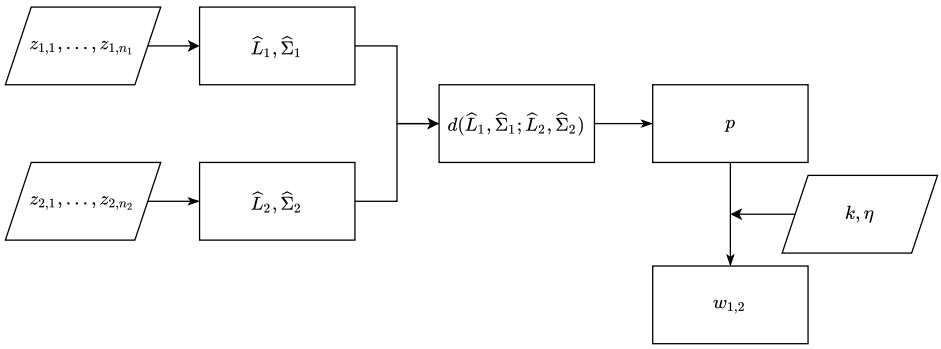


Figure 3: Procedure of transforming two samples into the weight used in the non-local mean filter.

3 Algorithm and Implementation Details

In this section, first, a description of the algorithm implemented and its main details are discussed. Then, the set of parameters used by the algorithm are explained in Subsection 3.4. Computational complexity is addressed in Subsection 3.5.

Figure 4 shows the flowchart of the non-local PolSAR filter. A more detailed description is provided in Algorithm 1.

Three are the main parts of the algorithm: Preprocessing, Filtering, and Post-processing.

3.1 Preprocessing

Non-local filters are highly computationally demanding because a large convolution mask must be computed for each pixel. This cost is largely increased for the case of PolSAR data, because each pixel value is a 9×9 complex matrix. These matrices must be inverted and their determinants calculated to estimate the stochastic distances, apart from other mathematical operations (statistical tests that involve transcendental functions). Also, the ENL (equivalent number of looks) is pixelwise estimated by the method discussed by Frery et al. [11]. Therefore, for a practical PolSAR non-local despeckling filter, it is necessary to implement the codes efficiently.

The input data consists of nine plain text files of size $m_{\text{rows}} \times n_{\text{columns}}$; one for each term of the covariance matrix (3). In the *Preprocessing* box shown in this figure, the covariance matrix for each pixel is estimated, and its inverse and the determinant are calculated and stored in memory. The estimates for each covariance matrix (see (5)) are obtained by a mean filter. This is done only once.

The methods to perform these tasks are in the `POLSAR_DAT.cpp` program:

- (line 118), `load_PolSAR_Data`: load the nine plain text files. If the user provides a reference image to estimate the image-quality indices (see Section 4), the nine plain text files are loaded (line 296, method `load_PolSAR_Data_Reference`). Both methods also show the Pauli representation of the uploaded data. The Pauli representation is explained below.

- (line 592), `preprocess_data`: from the uploaded data, the $\boldsymbol{Z}$ matrix (Equation (3)) for each pixel is built. Therefore, observations are transformed into a 3×3 matrix for each pixel within the Ω PolSAR image ($\Omega = \{\boldsymbol{Z}_{i,j} : 1 \leq i \leq m_{\text{rows}}, 1 \leq j \leq n_{\text{columns}}\}$). From each $\boldsymbol{Z}_{i,j}$, and by using a mean filter of suitable dimension (user-selected local patch size), the matrices, $\widehat{\boldsymbol{\Sigma}}_{i,j}$, for each image pixel i, j are calculated (Equation (5)). The inverses, $\widehat{\boldsymbol{\Sigma}}_{i,j}^{-1}$, and determinants, $|\widehat{\boldsymbol{\Sigma}}_{i,j}|$, $|\boldsymbol{Z}_{i,j}|$ are also calculated.

Notice that the first preprocessing step consists in expanding the image to deal with the edges. This is done by embedding the original image, whose support is the grid $S = \{1, \ldots, m_{\text{rows}}\} \times \{1, \ldots, n_{\text{columns}}\}$ in an image of support $S' = \{-(PW-1)/2, \ldots, 0, 1, \ldots, m_{\text{rows}}, m_{\text{rows}}+1, \ldots, m_{\text{rows}}+(PW-1)/2\} \times \{-(PW-1)/2, \ldots, 0, 1, \ldots, n_{\text{columns}}, n_{\text{columns}}+1, \ldots, n_{\text{columns}}+(PW-1)/2\}$. The new positions are filled with the values mirrored from the original image, e.g. the value at $(0, 0)$ is the same as at $(1, 1)$. This is done with the method `expand_image_outwards`, line 512.

Also, in the *Preprocessing* box, the ENL is obtained pixelwise by applying the bisection method to (6) to find the root within an interval centered into the nominal number of looks, L, which is a known value. The implementation of the bisection method is detailed in Algorithm 2. We adopted the usual approach for the limits of this interval: the minimum value is the minimum accepted value for the multilook data following the Wishart distribution, that is, 3. In contrast, the maximum value is twice the nominal value. This is done with the method `Estimate_ENL`, line 897.

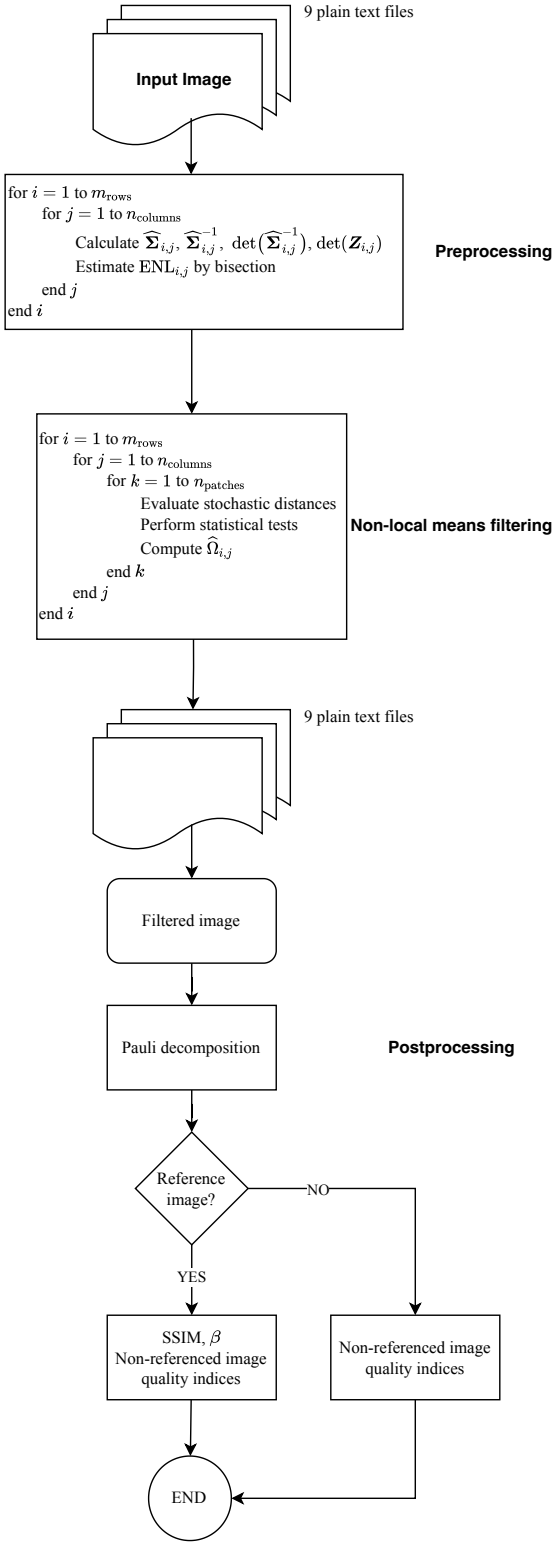


Figure 4: General flowchart of the non-local means approach. The non-referenced image quality indexes are the mean preservation, the variance reduction, and the equivalent number of looks (ENL).

Algorithm 1: Non-Local PolSAR Filter algorithm

Input : A fully PolSAR image (a folder name with the data), the nominal number of looks L, the size of the searching region SW, the size of the local patch PW, the confidence level η for the statistical test, the stochastic distance to use for estimating patches similarity, *distance_option*, the selection to estimate ENL locally ENL_option, the ROI coordinates to estimate image-quality indices xy, the reference image option GT, the edge detector to use *edge_detector_option*

Output: The filtered image $\widehat{\Omega}$ and the estimated image-quality indices (to a text file)

$\Omega_0 \leftarrow$ `load_PolSAR_Data(foldername)` *load data (nine text files of size $m_{rows} \times n_{columns}$)*

$\Omega \leftarrow$ `expand_image_outwards(`Ω_0`,` SW`,` PW`)` *image is mirror-reflecting across the borders*

$\Omega \leftarrow$ `preprocess_data(`Ω`)` *compute $\widehat{\mathbf{\Sigma}}$, $|\widehat{\mathbf{\Sigma}}|$, $\widehat{\mathbf{\Sigma}}^{-1}$, and $|\boldsymbol{Z}|$*

Initialize $ENL \leftarrow L$ *the ENL matrix is set to nominal L value*

if $ENL\_option == 1$ **then**

  ENL is pixelwise estimated (See Algorithm 2)

  $ENL \leftarrow$ `ENL_Estimate(L)`

Initialize $\widehat{\Omega} \leftarrow 0$

Non-Local Means filtering

foreach $i \leftarrow 1$ *to* m_{rows} **do**

  foreach $j \leftarrow 1$ *to* $n_{columns}$ **do**

    Initialize $w \leftarrow 0$ sum weights is set to 0

    foreach *patch k of size $PW \times PW$ in the $SW \times SW$ region centered at the position i, j* **do**

      if *distance_option* $== 1$ **then**

        $SD \leftarrow$ `Bhattacharyya_distance(i, j, k)` *Equation (12)*

      if *distance_option* $== 2$ **then**

        $SD \leftarrow$ `Hellinger_distance(i, j, k)` *Equations (9) and (12)*

      if *distance_option* $== 3$ **then**

        $SD \leftarrow$ `Kullback_Leibler_distance(i, j, k)` *Equation (11)*

      $p\_value \leftarrow$ `cdfchi(fabs(SD))` *χ^2 statistical test*

      $x \leftarrow \frac{p\_value - \eta/2}{\eta - \eta/2}$

      $w \leftarrow$ `smoother_function(x)` *transform p_value to a weight, Equation(1)*

      $n \leftarrow PW \times PW$

      $\widehat{\Omega}_{i,j} \leftarrow \frac{1}{\sum_{k=2}^{n} w(1,k)} \frac{1}{n} \sum_{k=2}^{n} w(1,k)\, Z_k$ *estimate the filtered value $\widehat{\Omega}_{i,j}$*

if $GT == 1$ **then**

  evaluate image-quality indices

  `SSIM_Estimates(`GT`,`$\widehat{\Omega}$`)` *there is a reference image GT: estimate SSIM*

  `Beta_Estimates(`$\widehat{\Omega}$`,` GT`,` *edge_detector_option*`)` *there is a reference image GT: estimate β*

`Estimates_in_ROIs(`xy`,` $\widehat{\Omega}$`)` *estimate referenceless metrics*

return $\widehat{\Omega}$ *return the filtered image $\widehat{\Omega}$*

Function *`SSIM_Estimates(`GT`,`$\widehat{\Omega}$`)`* *Equation(13)*

  return SSIM

Function *`Beta_Estimates(`$\widehat{\Omega}$`,` GT`,` edge_detector_option`)`* *Equation(14)*

  return β

Function *`Estimates_in_ROIs(`xy`,` $\widehat{\Omega}$`)`* *return estimates of the mean, variance, and ENL*

  return *estimates*

3.2 Filtering

With all the data ready for the estimation of the stochastic distance between patches, the usual overlapping non-local convolution is performed, and the central pixel (its related covariance matrix) is updated once the statistical test is done (see Algorithm 1). This is represented in Figure 4 by the small box which contains the routines to evaluate the stochastic distances and to perform the statistical test. The method used for these tasks (non-local convolution, evaluation of stochastic distance, statistical test, and central pixel updates) is `get_Sigma`, line 1017 for the case of using the nominal number of looks, L, or the method `get_Sigma_ENL`, line 1112, if the pixelwise estimated ENL is used.

As seen in Algorithm 1, the user can select among the three stochastic distances available: the Bhattacharyya distance, the Hellinger distance, and the Kullback-Leibler distance. Once again, two methods have been implemented for each of these distances depending on whether the nominal number of looks, L, or the pixelwise estimated ENL is used. For instance, for the Bhattacharyya distance, the method used when the nominal L values applies is `Bhattacharyya_distance_ENL`, line 1095, and the method `Bhattacharyya_distance_L1_L2`, line 1239 is used for the other case. The same applies for the other two stochastic distances.

It is important to indicate that the codes are implemented in C++ and the unique class PolSAR_Data (see header file, "PolSAR_Data.h"), is defined, and a unique object is declared, PolSAR_Image (`main.cpp`, line 306). By using simple "set" methods, variables are assigned to the object, and they are *naturally* available for methods related to the owner class. The "get" methods allow to get variables from the class (see, for instance, in this header file, line 111 or line 229). Due to that, it is not necessary to pass all variables used in the methods as arguments. This is, for instance, the case of the call to the method `ENL_Estimate` detailed in Algorithm 2. Although this method requires the precalculated $|\boldsymbol{Z}|$, $|\widehat{\boldsymbol{\Sigma}}|$, $\widehat{\boldsymbol{\Sigma}}^{-1}$, only the L nominal value is passed as argument.

3.3 Postprocessing

For a PolSAR filtering operation, 9 files are obtained (one for each band). The Pauli representation is the most commonly accepted result for visual evaluation. This color codification provides a well-suited optical-like image [18] and is generated from the diagonal terms of the covariance data (HH, HV and VV bands). Figure 10 shows an example of such codification.

Additionally, to account for the vast dynamic range of PolSAR data, the Pauli representation works with the logarithm of the data (decibels), and histogram equalization is also required (see the methods `double_dB_char` and `image_adjust` in the file `Auxilary_methods.cpp`, lines 66 and 140, respectively).

This global description of the algorithm concludes with evaluating the metrics to assess the quality of the despeckling operation done on the original PolSAR data. Several image-quality indices participate in this assessment, some referenced and others non-referenced. These measures of quality are described in Section 4. One of the referenced metrics requires an edge detector for evaluating edge preservation. For that purpose, a standard Canny method has been included in the codes (see the method `CannyEdges` in the file `Edge_detectors.cpp`, line 144). The detection of edges through the Laplacian operator and the zero-crossing algorithm are also available (see in the same file, the method `edges`, line 376). This edge detector is also used for estimating the β metric (see Section 4).

More details of the non-local PolSAR filter implementation are in the Appendix.

3.4 Choice of Parameters

The parameters that define an NLM filter for PolSAR data are user-defined. The default setup is:

Algorithm 2: Method to estimate ENL by the bisection algorithm

Input : precomputed $|\boldsymbol{Z}|$, $|\widehat{\boldsymbol{\Sigma}}|$, $\widehat{\boldsymbol{\Sigma}}^{-1}$, the nominal number of looks L
Output: The ENL matrix
Initialize $tolerance \leftarrow 1.0E-4$ — *the tolerance value to finish the algorithm*
Initialize $n_{\max} \leftarrow 100$ — *the maximum number of iterations allowed*
Initialize $ENL \leftarrow L$ — *the ENL matrix is set to the nominal value L; its size is* $m_{rows} \times n_{columns}$
foreach $i \leftarrow 1$ *to* m_{rows} **do**
  foreach $j \leftarrow 1$ *to* $n_{columns}$ **do**
    Initialize $a \leftarrow 3$ — *PolSAR valid minimum number of looks*
    Initialize $b \leftarrow 2L$ — *maximum number of looks to search for*
    Initialize $n \leftarrow 1$ — *n is the counter of iterations*
    rest $\leftarrow \log(|\boldsymbol{Z}_{i,j}| \,/\, |\widehat{\boldsymbol{\Sigma}}_{i,j}|) - \operatorname{trace}(\widehat{\boldsymbol{\Sigma}}_{i,j}^{-1} \cdot \boldsymbol{Z}_{i,j})$ — *constant term in Equation (6)*
    $f_a \leftarrow$ `ENL_function(a, rest)` — *evaluate the function at endpoint a*
    $f_b \leftarrow$ `ENL_function(b, rest)` — *evaluate the function at endpoint b*
    if $sign(f_a) == sign(f_b)$ **then**
      continue — *no root in [a,b]; proceed to next i, j pixel*
    while $n \leq n_{\max}$ **do**
      $c \leftarrow (a+b)/2$ — *central point of interval (a, b)*
      $f \leftarrow$ `ENL_function(c, rest)` — *call to the function, Equation (6)*
      if $(|f| < tolerance$ ***or*** $|b-a| < tolerance)$ **then**
        $ENL_{i,j} \leftarrow c$ — *root found*
        break — *it proceeds to next i, j pixel*
      if $sign(f) == sign(f_a)$ **then**
        $a \leftarrow c$ — *root lies in* (c, b)
        $f_a \leftarrow f$
      else
        $b \leftarrow c$ — *root lies in* (a, c)
        $f_b \leftarrow f$
      $n \leftarrow n+1$ — *next iteration*
return ENL — *estimated* $\widehat{ENL}$ *matrix*
Function *ENL_function(L, rest)* — *Equation(6)*
  Initialize $sum \leftarrow 0$
  foreach $i \in \{0, 1, 2\}$ **do**
    if $(L-i) > 0$ **then**
      $sum \leftarrow sum + \psi^{(0)}(L-i)$
  return $(3 \cdot (\log(L)+1)) + rest - sum)$ — *return the root*

- SW (size of the searching window): a large SW value implies higher computational cost. From the experiments performed, we set SW to 7 and 11.
- PW (size of the patch): although the computational cost is not dramatically related to the size of the patch, we set PW to 3 and 7. We impose the constraint $SW > PW$.
- η value (the confidence level in (1)): used in the statistical test and weight transforming. This parameter is responsible for the number of patches considered similar to the central pixel to be estimated. The larger η, the fewer patches participate in the final estimation. In our implementation, η is set to 0.8, 0.9, and 0.99 to explore its effect on the quality of results. Note that η controls the acceptance or rejection of patches, and so, it is the more decisive parameter.
- SD (stochastic distance): three options are available, namely the Kullback-Leibler, Bhat-

tacharyya, and Hellinger distances.

As shown in Table 2, the user can select different parameters either directly from the online-interface or through command line execution[2].

Parameters	Default Value	Interface?	Accepted values
Search window size (SW)	7	✓	$7, 9, 11$
Patch size (PW)	3	✓	$3, 5, 7, 9, 11$
η value	0.8	✓	$(0.0, 1.0)$
Stochastic distance (SD)	KL	✓	KL, Hell, Bhatta

Table 2: Parameters for the non-local means POLSAR filter.

For the offline version, the user can also select:

- to estimate ENL pixelwise (default option), or use a fixed value for the whole data (for instance, the known nominal number of looks L),
- to use the original mapping function by Torres et al. [30], or the proposed smoother function (default option),
- to detect edges by using the Canny method (default option), or the zero crossings of the Laplacian of a Gaussian.

Local estimation of ENL enhances the model expressiveness, but also the computational cost. It is worth mentioning that the Wishart model is adequate for fully developed speckle and that the hypotheses under which it is valid may not be met, e.g., by data from forests and urban areas. Allowing ENL to vary diminishes the discrepancy between the observations and the Wishart law.

3.5 Computational Complexity

As detailed above, all the elements that can be pre-computed off the algorithm's main loop are pre-calculated. Thanks to this precaution, the computational complexity is the usual for non-local filters: For an image of size $|\Omega|$ and size of search window SW, the maximum complexity is of the order of $O(|\Omega|\, SW^2)$. Such complexity does not depend on the size of the local patch (PW) since all calculations needed for the estimation of stochastic distances are done before the main loop.

Note that, in our implementation, no iteration is used, and no pixel pre-selection is done. Pixel pre-selection would slightly reduce the computational cost by using only the similar patches of larger p-value to estimate the central pixel.

The code runs in parallel. The processing of the synthetic image used in the experiments (image size $|\Omega| = 500 \times 500$ pixels), on Intel© Core® i7-7700HQ CPU@ 2.8 GHz, 16 GB RAM, with 4 cores and hyperthreading (8 virtual cores) takes ≈ 20 s (including pre-processing and image generation of final results).

Tables 3, 4 and 5 report the computational cost of the experiments performed in Section 5.

4 Measures of Quality

Argenti et al. [1] discuss several measures of speckle filter quality. They can be categorized as with-reference indexes, i.e., when the "perfect" or "noiseless" image is available, and without-reference indexes (the most usual case in practice). We use the following metrics:

[2]In Table 2, KL: Kullback-Leibler, Hell: Hellinger and Bhatta: Bhattacharyya.

- MSSIM (Mean Structural Similarity Index), which measures the similarity between the original and despeckled images with local statistics (mean, variance and covariance between the unfiltered and despeckled pixel values). This measure is bounded in $(-1, 1)$, and a good similarity produces value close to 1. MSSIM is usually denoted as SSIM, and we adopt this notation. It is to easy implement through its mathematical definition

$$\mathrm{SSIM}(x, y) = \frac{\left(2\widehat{\mu}_x\widehat{\mu}_y + c_1\right)\left(2\widehat{\sigma_{xy}} + c_2\right)}{\left(\widehat{\mu}_x^2 + \widehat{\mu}_y^2 + c_1\right)\left(\widehat{\sigma}_x^2 + \widehat{\sigma}_y^2 + c_2\right)}, \tag{13}$$

  where x and y are selected areas of size $m \times n$ within the image I (reference image) and the image $\widehat{I}$ (filtered image) respectively. $\widehat{\mu}_x$, $\widehat{\mu}_y$, $\widehat{\sigma}_x^2$, $\widehat{\sigma}_y^2$ are the sample mean and the sample variance estimated in the areas x and y respectively, and $\widehat{\sigma}_{xy}$ is the covariance of x and y. The variables $c_1 = (k_1 D)^2$ and $c_2 = (k_2 D)^2$, where D is the dynamic range of the image[3], and $k_1 = 0.01$ and $k_2 = 0.03$ avoid division by zero. SSIM can be measured in the whole image or in several selected areas $(x_i,\ y_i)$ and then averaged to get the MSSIM (Structural Similarity Index) metric. SSIM is estimated with the method `SSIM_Estimates`, located in the file `Estimators.cpp`, line 172.

- β assesses edge preservation by measuring the correlation between edges in the reference and denoised images. Edges are detected either by the Laplacian or the Canny filter (both methods are available for the user in the released code, although we present only results with the latter). The β index ranges between 0 and 1, and the bigger it is, the better the filter is (the ideal edge preservation yields $\beta = 1$). Once the edges of both images, the original noisy (or reference image), I and the despeckled one, $\widehat{I}$, are available, β is obtained through

$$\beta = \frac{\Gamma\left(\Delta I - \overline{\Delta I}, \widehat{\Delta I} - \overline{\widehat{\Delta I}}\right)}{\sqrt{\Gamma\left(\Delta I - \overline{\Delta I}, \Delta I - \overline{\Delta I}\right)}\sqrt{\Gamma\left(\widehat{\Delta I} - \overline{\widehat{\Delta I}}, \widehat{\Delta I} - \overline{\widehat{\Delta I}}\right)}}, \tag{14}$$

  where $\Gamma(I_1, I_2)$ is given by

$$\Gamma\left(I_1, I_2\right) = \sum_{i=1}^{K} I_{1_i} \cdot I_{2_i}. \tag{15}$$

  ΔI and $\widehat{\Delta I}$ are the high-pass filtered versions of images I and $\hat{I}$, respectively, obtained with a sliding Laplacian pixel kernel window of size 3×3 or another edge detector such as the Canny detector; $\overline{\Delta I}$ and $\overline{\widehat{\Delta I}}$ are the average values of the image I and the average of the high-pass filtered version of the image $\widehat{\Delta I}$, respectively. This metric evaluates the correlation between the ground truth edges within the original image and the edges in the denoised image detected by means of the Laplacian filter (or the Canny filter). β is estimated with the method `beta_estimator`, located in the file `Estimators.cpp`, line 68, and by default, the Laplacian filter is used.

- Mean preservation: the mean value of the noisy image μ_N estimated within a large enough homogeneous area (same radar signature) is compared to the mean value for the denoised image, μ_D. A good filtering operation should preserve that value ($\mu_N \approx \mu_D$).

- Variance reduction: the variance of the filtered image, var_D must be smaller than the variance of the noisy image (var_N), both measured in a large enough textureless area.

[3]For an 8 bit resolution image, i.e., an image with 255 possible grayscale values, $D = 2^8 - 1 = 255$.

- ENL (Equivalent Number of Looks), is among the simplest and most spread measures of quality of despeckling filters. As explained above, it can be obtained by solving (6), although, as a non-referenced metric, it is usually estimated, in textureless areas and intensity format, as the ratio of the squared sample mean to the sample variance, i.e., the reciprocal of the squared coefficient of variation. ENL is proportional to the signal-to-noise ratio. The higher ENL is, the better the quality of the image is in terms of speckle reduction. Much care must be taken when interpreting ENL; it is well known that large ENL values are easily obtained just by over filtering an image, which severely degrades details and gives the filtered image an undesirable blurred appearance. These metrics are evaluated with the method `Estimates_in_ROIs`, located in the file `Estimators.cpp`, line 275.

SSIM and β are widely applied in SAR images and natural images, and a reference image is needed for their evaluation. The last three metrics (mean preservation, variance reduction, and ENL) need no such reference image.

All the estimators are implemented in the file `Estimators.cpp`. The edge detectors (the Canny and the Laplacian zero crossing filter) are coded in the file `Edge_detectors.cpp`. The Canny method requires setting two threshold parameters (the Low and High threshold), which control the number of detected edges. The default values used in the implementation are in the `Edge_detectors.cpp` file (lines 69 to 72). Those values work well for all the cases tested, but if not, they must be suitably modified by the user. That is why one of the program outputs shows the map of detected edges, so the user can check if the threshold values provide acceptable edge detection.

The usual approach for evaluating the performance of a PolSAR filter, as many bands are involved with PolSAR data, consists of estimating all the metrics (SSIM, β, mean preservation, variance reduction and ENL) for the main diagonal components of the covariance matrix (HH, HV and VV bands), and averaging these values. This is the approach followed in this paper. Additionally, it is also common to estimate those metrics on what is known as the "span image", which is obtained by averaging those components and producing a single band and then estimating all the metrics on that span image. This is also done in this work.

Apart from that well-known metrics, in SAR and PolSAR images, the evaluation of the performances is completed by a visual inspection (by an expert) of the filtered data (the diagonal terms and the span image) to assess the preservation of edges and fine details.

5 Results

An extensive analysis of the performances of the non-local PolSAR filter can be easily done through the user-friendly *demo*. In this section, we focus on showing how the setting of the η value, the use of the available stochastic distances, and the size of the patches modify the quality of the results. We want to illustrate that the *demo* allows tuning the filter for a given input data easily.

We follow the usual procedure for testing filters: first, the filter is extensively tested on a synthetic image with an available reference (ground-truth). In doing so, conclusions can be achieved utilizing numerical (objective) image quality indices. Such indices are also discussed in the following. Then, the performance of the filter is evaluated on actual PolSAR data. In both cases, our interest relies on denoising the images while preserving details.

The above experimental setup for the case of the synthetic data can be enriched by running Monte Carlo experiments [21]. However, due to time constraints in the execution times related to IPOL's online demo, those experiments have not been included in this paper. We remark that the user can easily prepare Monte Carlo experiments from the available codes.

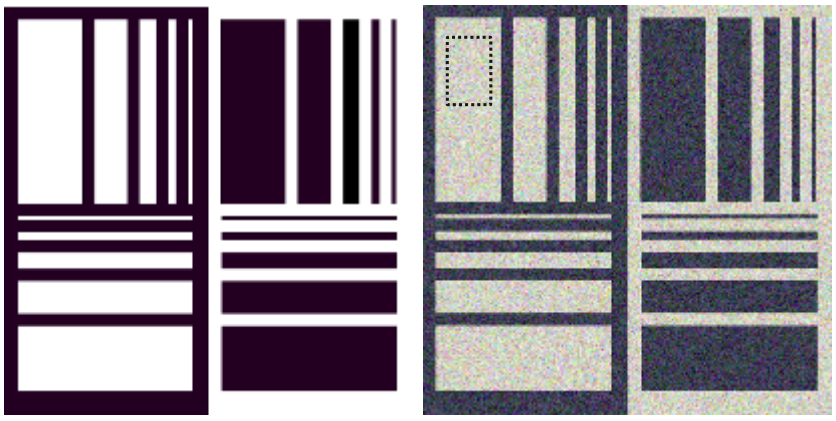
5.1 Simulated Data

We use the simulated data illustrated in Figure 5. This image contains large homogeneous areas and many edges and, thus, is helpful to test the filter's performance at reducing noise while preserving fine details. The images have 500×500 pixels. To generate a fully PolSAR sample, we use two covariance matrices from actual PolSAR data and a Wishart matrix generator that can produce samples with a given number of looks. The covariance matrices used for the simulation are

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 9.6289 & 0.1917 - j0.0358 & -1.5464 + j1.9139 \\ 0.1917 + j0.0358 & 0.5671 & -0.0580 + j0.1681 \\ -1.5464 - j1.9139 & -0.0580 - j0.1681 & 4.7225 \end{bmatrix} 10^5, \text{ and}$$

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 3.2556 & 0.0556 + j0.0787 & 2.4046 - j2.7287 \\ 0.0556 - j0.0787 & 0.1647 & -0.0146 - j0.0482 \\ 2.4046 + j2.7287 & -0.0146 + j0.0482 & 6.1028 \end{bmatrix} 10^4,$$

which were observed in urban and pasture regions [11].



(a) Reference image

(b) Pauli codification of the three-looks simulated sample

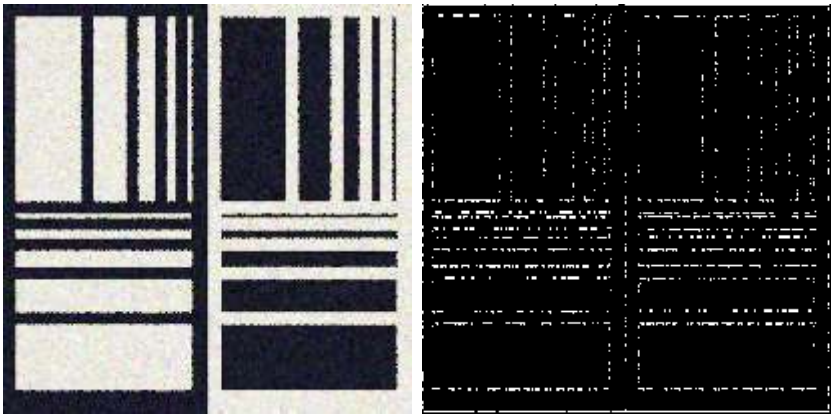
Figure 5: Simulated PolSAR data used in the experiments. The black rectangle in dotted line is the selected area to estimate the non referenced metrics.

Figure 6 shows results, in Pauli representation, obtained by using different setting parameters and two stochastic distances (Kullback-Leibler and Bhattacharyya) for the PolSAR simulated data. Detected edges by using the Canny detector, for the filtered data (span data), are also depicted. Similar results are shown in Figure 7 by using different parameters and the Hellinger stochastic distance.

As seen in these figures, the effect of speckle diminishes with the increase of size patches. However, although reasonably well preserved, edges also degrade with the increase of patch sizes. Note that the results (for a given parameter setting) are visually similar. An objective evaluation is provided by the metrics detailed in the previous section and shown in Figure 8 (metrics evaluated within the large homogeneous area), and Figure 9 (metrics accounting for the whole image).

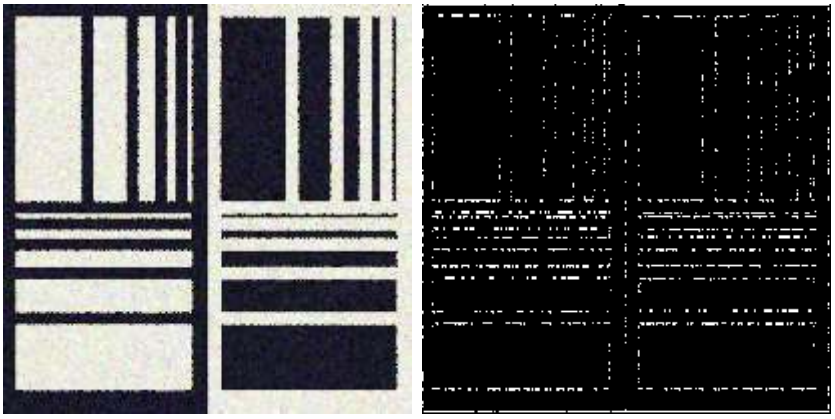
Several conclusions can be derived from these plots. First, we analyze the local estimators (measured in a large homogeneous area).

- mean preservation (μ_p): a maximum divergence of 5 % from the original value (measured in the noisy data) is commonly accepted as a good result. In this case, the values are considerably smaller ($|\mu_p| < 0.5$), so, acceptable. There are noticeable differences for the stochastic distances



(a) Filtered result by using the Kullback-Leibler stochastic distance

(b) Detected edges for the Kullback-Leibler stochastic distance.



(c) Filtered result by using the Bhattacharyya stochastic distance

(d) Detected edges for the Bhattacharyya stochastic distance.

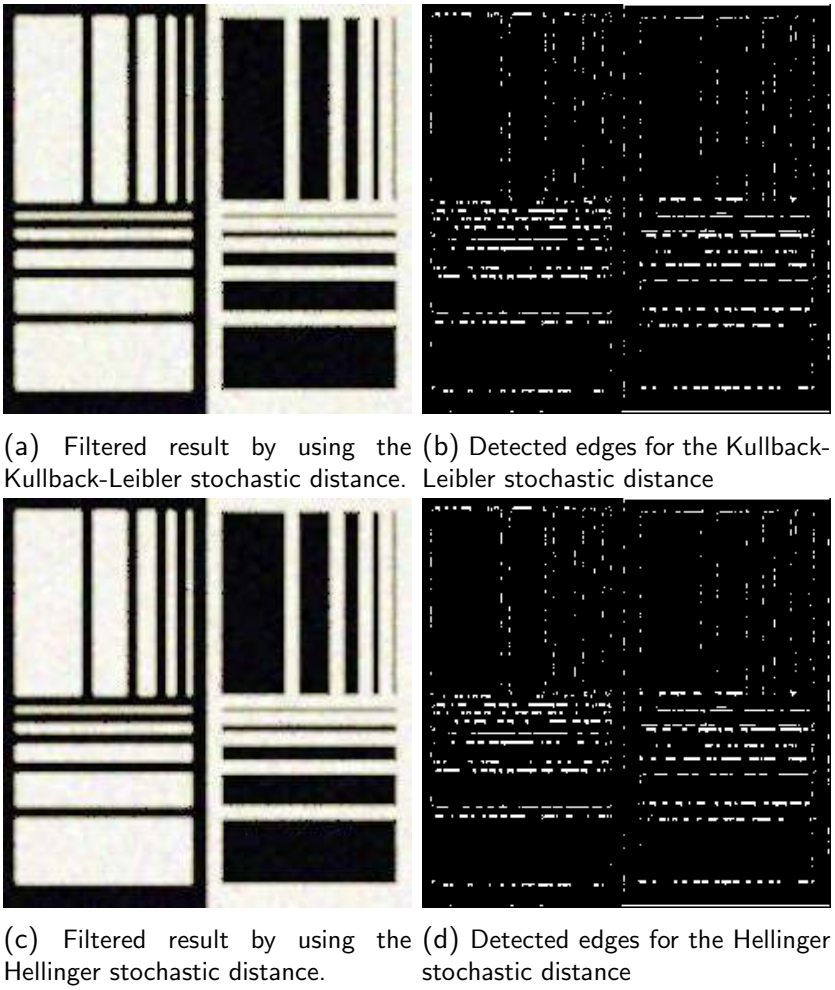
Figure 6: Filtered results for the setting: 7×7 (3×3), $\eta = 0.8$ and two stochastic distances.

and the p-value, although not big, the mean is better preserved for a small patch size. The best result is for the Hellinger distance with p-value equal to 0.8 and 7×7 patch size, although, as mentioned above, all results are excellent. This is the expected result from a robust non-local filter.

- Variance reduction or standard deviation reduction (σ_p as shown): non-local filters perform well on strongly reducing the speckle, with a final value, for the worst case, of a reduction $\approx -90\,\%$. Larger patches render better variance reduction (minor standard deviation values, close to $-100\,\%$). Note that dependence on the p-value is irrelevant when using large patches; that is, no more reduction is possible (a saturated result). A similar *best* variance reduction is obtained with the three distances for the large patch case. For the 7×7 patch size, the best result is, as above, for the Hellinger distance.

- ENL: as explained above, a large ENL value relates to a reduction of speckle. As seen in Figure 8, ENL gets a large relative percentage improvement value (larger than 5000) for any combination of parameters for the case of using large patches (11×11). This is also the expected result because larger patches mean that more pixels participate in estimating the weights to update the central pixel. Note also that an increase of the p-value reduces the ENL value. This is because fewer patches are selected to update the central pixel.

For the other estimators (measured on the whole image),



(a) Filtered result by using the Kullback-Leibler stochastic distance.

(b) Detected edges for the Kullback-Leibler stochastic distance

(c) Filtered result by using the Hellinger stochastic distance.

(d) Detected edges for the Hellinger stochastic distance

Figure 7: Filtered results for the setting: 11×11 (7×7), $\eta = 0.99$ and two stochastic distances.

- SSIM: in this case, it is clear that the Hellinger and the Bhattacharyya distances provide the best results. The Kullbak-Leibler distance indeed provides poor results for small p-values. This is an interesting result and clearly shows the benefits of a simple analysis using the *demo* to set the filter parameters better.

- β: edge preservation is similar for all the settings. Also, edges degrade with the size of the patches (it is worse for larger patches).

From the above analysis, and taking into account that we are discussing the worst (noisiest) possible case (with ENL set to 3), we can deduce that the filter performs well, as expected, in all the cases, and that the values of SSIM and β estimators show opposite behavior. This means that one gets a better SSIM to the price of degraded edges (worse β). Additionally, these conclusions are not trivially deduced from the visual results, cf. Figure 6 and Figure 7, but may impact the performance of subsequent procedures, like classification or segmentation.

Table 3 presents the running times for the experiments. The execution time using the default patches configuration (7×7 and 3×3) is $\approx 10\,\mathrm{s}$ for the main loop (convolution and estimation of the stochastic distances). The computation time is around twice this value for the whole execution (within parentheses in the table), that is, including the preprocessing stage for calculating matrix operations (assembling the matrices, inversion, and calculus of determinants) and the post-processing stage to generate all images, output files and evaluating the referenced and non referenced estimators. There are no significant differences in the choices of either the stochastic distances or the selected η

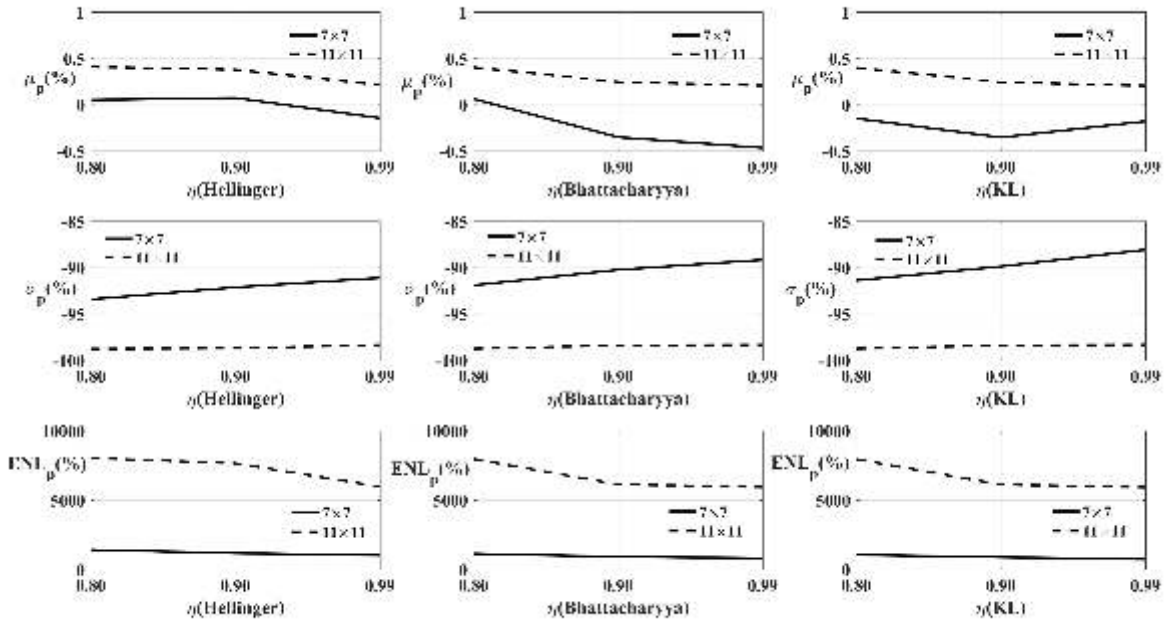


Figure 8: Some results for the simulated PolSAR data for the measures in the selected homogeneous area (relative values respected to the original ones and in percentage).

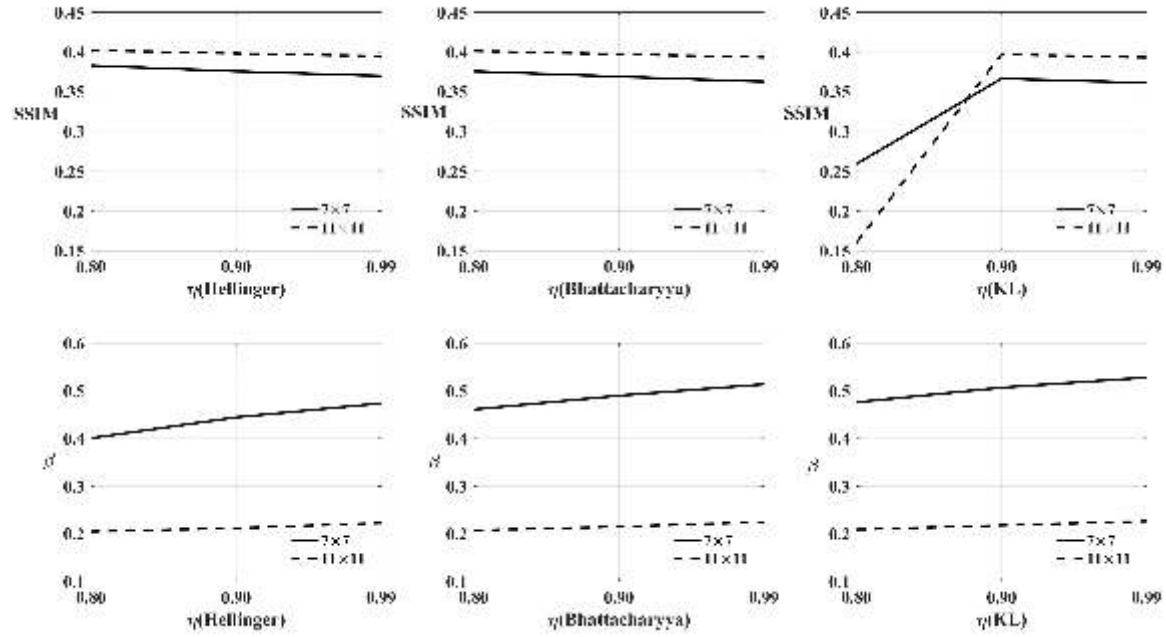


Figure 9: Some results for the simulated PolSAR data for the referenced estimators β and SSIM.

values. The running time for the patches configuration 11×11 and 5×5 is $\approx 30\,\mathrm{s}$, and less than $40\,\mathrm{s}$ for the whole execution.

It is interesting to note that the running time is related to the loops to manage the large and local patches and the estimation of the stochastic distances. Although the expressions for the three distances are different, they require similar computing times. These similar (and affordable) times result from careful coding: as much information as possible is obtained during the preprocessing

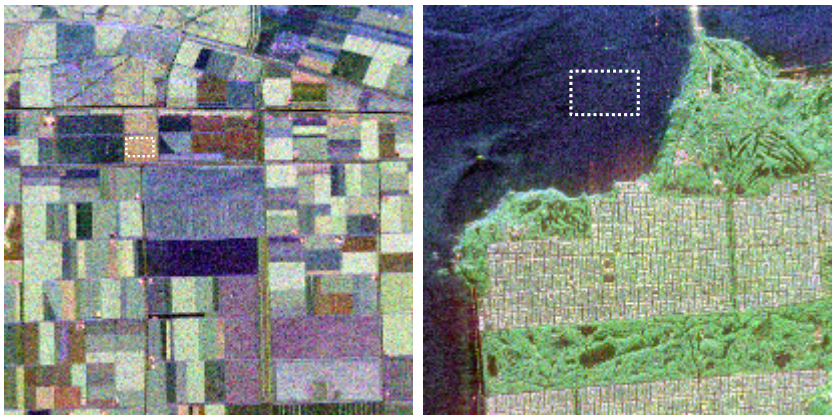
Hellinger distance	$\eta = 0.8$	$\eta = 0.9$	$\eta = 0.99$
$7 \times 7, 3 \times 3$	11.33 (20.07)	11.44 (19.96)	10.93 (19.24)
$11 \times 11, 5 \times 5$	27.46 (36.00)	28.67 (37.20)	29.11 (37.82)
Bhattacharyya distance	$\eta = 0.8$	$\eta = 0.9$	$\eta = 0.99$
$7 \times 7, 3 \times 3$	10.79 (19.20)	10.68 (19.04)	10.79 (19.60)
$11 \times 11, 5 \times 5$	27.31 (35.79)	28.81 (37.47)	27.66 (36.79)
Kullback-Leibler distance	$\eta = 0.8$	$\eta = 0.9$	$\eta = 0.99$
$7 \times 7, 3 \times 3$	10.73 (19.09)	10.77 (19.34)	11.02 (19.51)
$11 \times 11, 5 \times 5$	28.10 (36.56)	28.04 (36.61)	28.64 (37.19)

Table 3: Computational times for the simulated PolSAR data (seconds).

stage and then remains unaltered. Note also that approximately 10 s (regardless of the patches configuration) of the running time are related to the pre- and post-processing of the data.

5.2 Actual Data

We tested the non-local filters on the two fully PolSAR data shown, in Pauli decomposition, in Figure 10. Flevoland is a four-look 1024×750 pixels image over a region in the Netherlands. The San Francisco image (900×1024 pixels) also has four nominal looks. The AIRSAR sensor obtained both images in L-band with 10 m $\times$ 10 m of spatial resolution. We worked with a sub-image of size 500×500 for the experiments shown in this section.



(a) Flevoland image (b) San Francisco image

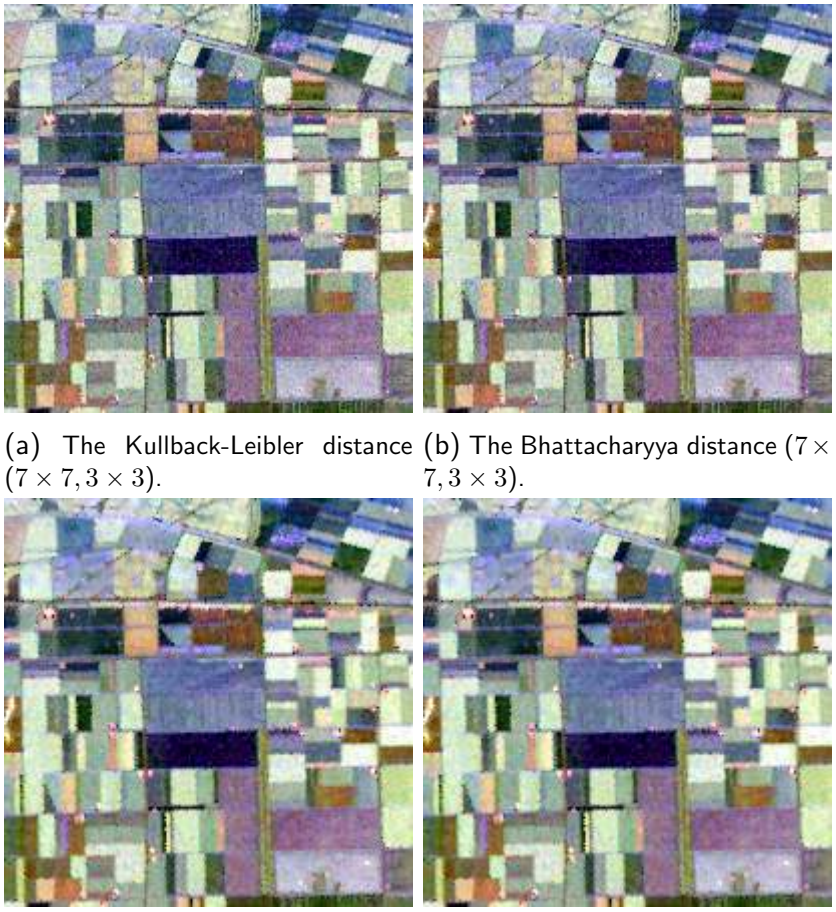
Figure 10: Original images from Flevoland and San Francisco in Pauli decomposition. The white rectangle in dotted line is the selected area to estimate the non referenced metrics.

As no reference image is available, the metrics used are the mean preservation, the variance reduction, and the estimation of the number of looks measured in a homogeneous area (white rectangle within the shown images).

Visual results for both PolSAR data (see Figures 11 and 12) show, for the different settings of filter parameters, that speckle has been strongly reduced with acceptable edge and fine details preservation. Strong scatterers (bright isolated pixels) remain unaltered, as expected from a competitive SAR/PolSAR despeckling filter. Such bright pixels are related to complex signal scattered effects from reflection from manufactured structures like buildings, vessels, cars, and fences. Also, as the

patch size increases (11×11), less speckle content remains, but the filtered image appears slightly blurred.

Analogously for the simulated data, it is difficult to extract conclusions from the mere visual inspection of the filtered results. The non referenced numerical estimators are shown in Figure 13 for the Flevoland image and Figure 14 for the San Francisco data, respectively.



(a) The Kullback-Leibler distance ($7 \times 7, 3 \times 3$).

(b) The Bhattacharyya distance ($7 \times 7, 3 \times 3$).

(c) The Kullback-Leibler distance ($11 \times 11, 5 \times 5$).

(d) The Hellinger distance ($11 \times 11, 5 \times 5$).

Figure 11: Filtered results for Flevoland for $\eta = 0.8$.

Results are, in general, similar for both images (same trend and range of variation for σ_p and ENL). However, there are noticeable differences with the simulated data. In particular, the ENL improvement is inferior (< 1000) for the simulated (5000), and it is less dependent on the p-value. The mean preservation has larger values than with simulated data. However, they are acceptable for all the configurations: ($|\mu_p| < 5\,\%$) for Flevoland and inferior with the San Francisco data ($|\mu_p| < 0.5\,\%$).

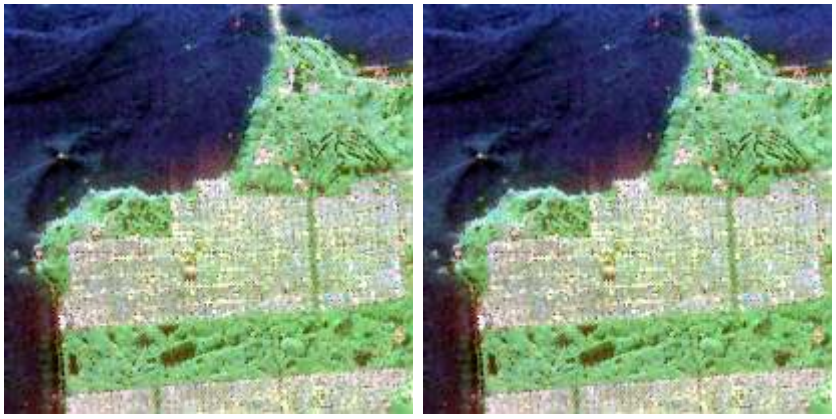
This is also an expected result, and it keeps open the debate about the convenience of extrapolating conclusions from simulated data to the complex scenario of actual PolSAR data.

Therefore, it seems clear that no general setting is valid for all data and, consequently, possibilities like the ones offered by the *demo* are of great practical use.

The running times for the experiments for both actual data can be seen in Table 4 and Table 5. Similar conclusions than for the simulated PolSAR data case can be derived. The computation time for using the patches configuration 7×7 and 3×3 is $\approx 10\,\mathrm{s}$ (twice for the whole execution) and for the patches configuration 11×11 and 5×5 is less than $40\,\mathrm{s}$. Again, we verified no significant differences among stochastic distances or the selected η values.

(a) The Kullback-Leibler distance $(7 \times 7, 3 \times 3)$.

(b) The Bhattacharyya distance $(7 \times 7, 3 \times 3)$.



(c) The Kullback-Leibler distance $(11 \times 11, 5 \times 5)$.

(d) The Hellinger distance $(11 \times 11, 5 \times 5)$.

Figure 12: Filtered results for San Francisco for $\eta = 0.99$.

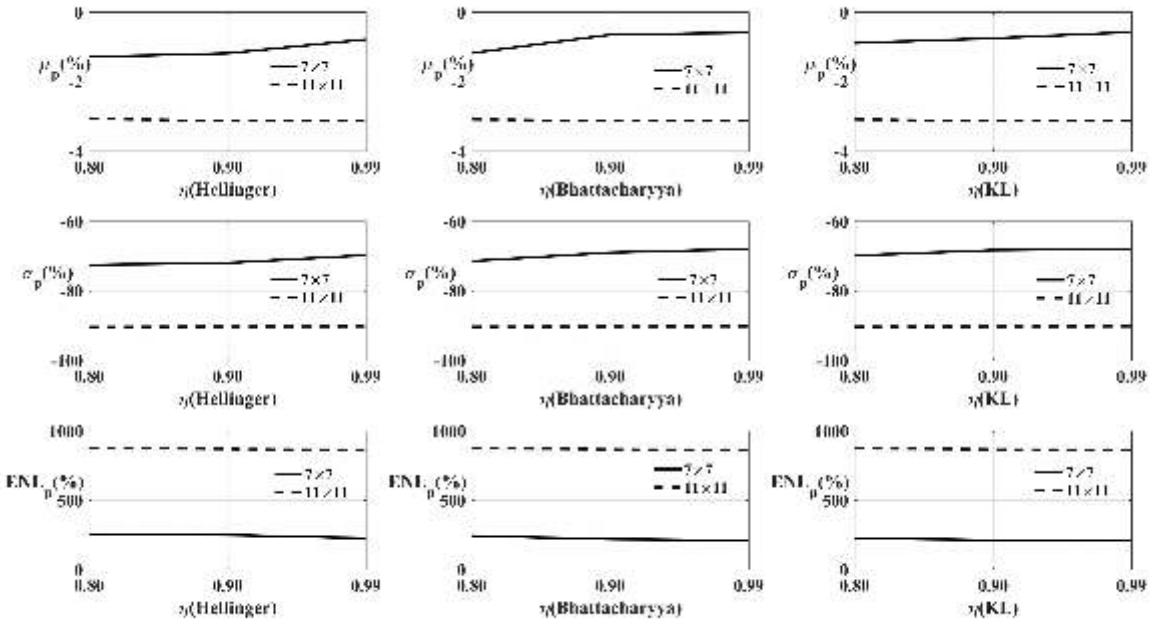


Figure 13: Some results for Flevoland PolSAR data for the non referenced estimators.

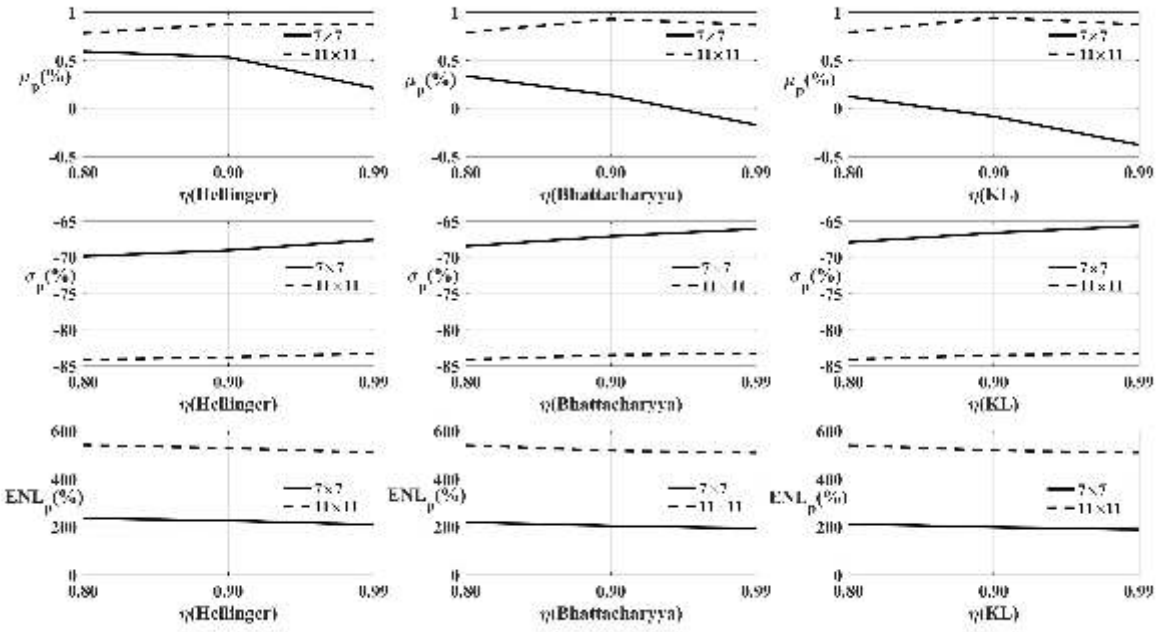


Figure 14: Some results for San Francisco PolSAR data for the non referenced estimators.

Hellinger distance	$\eta = 0.8$	$\eta = 0.9$	$\eta = 0.99$
$7 \times 7, 3 \times 3$	10.63 (21.60)	10.71 (20.17)	10.59 (20.09)
$11 \times 11, 5 \times 5$	26.53 (37.15)	26.71 (36.38)	27.25 (36.90)
Bhattacharyya distance	$\eta = 0.8$	$\eta = 0.9$	$\eta = 0.99$
$7 \times 7, 3 \times 3$	10.62 (20.10)	10.46 (19.87)	10.56 (21.18)
$11 \times 11, 5 \times 5$	26.51 (36.10)	26.25 (35.90)	26.20 (37.48)
Kullback-Leibler distance	$\eta = 0.8$	$\eta = 0.9$	$\eta = 0.99$
$7 \times 7, 3 \times 3$	10.84 (20.43)	10.99 (20.47)	10.55 (20.05)
$11 \times 11, 5 \times 5$	26.69 (36.20)	27.77 (37.38)	27.12 (36.82)

Table 4: Computational times for Flevoland data (seconds).

Hellinger distance	$\eta = 0.8$	$\eta = 0.9$	$\eta = 0.99$
$7 \times 7, 3 \times 3$	11.06 (21.28)	12.69 (22.43)	11.20 (21.00)
$11 \times 11, 5 \times 5$	26.70 (36.22)	26.88 (36.36)	27.27 (36.70)
Bhattacharyya distance	$\eta = 0.8$	$\eta = 0.9$	$\eta = 0.99$
$7 \times 7, 3 \times 3$	10.95 (20.40)	10.79 (20.15)	10.57 (19.83)
$11 \times 11, 5 \times 5$	27.53 (37.01)	26.88 (36.32)	26.81 (36.30)
Kullback-Leibler distance	$\eta = 0.8$	$\eta = 0.9$	$\eta = 0.99$
$7 \times 7, 3 \times 3$	10.73 (20.09)	11.57 (21.04)	10.83 (20.12)
$11 \times 11, 5 \times 5$	27.12 (35.57)	27.41 (36.92)	27.40 (36.78)

Table 5: Computational times for San Francisco data (seconds).

6 Conclusions

PolSAR images provide extremely rich remote sensing information. However, since they use coherent radar signals for the image acquisition, data come corrupted with multiplicative noise known as speckle. Such speckle content must be reduced through efficient filters for a correct interpretation of the data and successful post-processing. This paper presented a tool for exploring non-local despeckling filters for PolSAR data based on statistical tests that rely on stochastic distances.

To minimize computational cost, we provide an efficient implementation that explores pre-calculating and storing in memory as much information as possible to avoid multiple repetitive matrix operations. The result is an efficient code that allows the user to explore each filter setup's properties and find the optimal tuning for a given PolSAR data set.

Appendix

This appendix provides details about the code and the results generated by the program.

- Input data:
  - The fully covariance PolSAR data consists of 9 plain texts, with the usual names, `C11.txt` (the term C_{11} in matrix format), `C12_real.txt` ($\Re(C_{12})$), `C12_ima.txt` ($\Im(C_{12})$), `C13_real.txt` ($\Re(C_{13})$), `C13_ima.txt` ($\Im(C_{13})$), `C22.txt` (C_{22}), `C23_real.txt` $\Re(C_{23})$), `C23_ima.txt` ($\Im(C_{13})$) `and C33.txt`. Each file must contain a header (first line) indicating the number of rows and columns as seen in Figure 15.
  - Reference image: if a reference image (ground-truth) is provided, the program will get measures for the referenced metrics (the beta estimator and the SSIM index) accordingly. Note that such reference image is indeed a PolSAR data also, so, 9 plain files (covariance data) are required.
- Output data: Several data are generated.
  - Filtered data: 9 plain files (covariance data) with the same format as the PolSAR input files (see Figure 15).
  - Metrics data: they are collected in the file `metrics.txt`, also a plain file. All the referenced metrics are calculated if the user provides a reference image (the beta and the SSIM estimators). If not, only the non-referenced metrics are estimated. The file also indicates the edge detector used (the Canny method or the simple Laplacian zero-crossing method). See in Figure 16 a complete metrics file as an example.
  - Estimated ENL: this plain text contains the ENL (Equivalent Number of Looks) estimates for the PolSAR data. The values are obtained numerically for each pixel from the covariance matrix. Figure 17 shows the values estimated for Flevoland data ($L = 4$).
  - Visual representation: the images from the original noisy data and the filtered result for each band (HH, HV and VV), the Pauli representation for the original noisy data, and the filtered result are generated and saved in TIFF (Tagged Image File Format) format. Figure 18 shows all the images generated for the Flevoland case.
  - Non-Hermitian pixels: a binary image showing those pixels whose covariance matrix is not positive definite. This helps to understand the obtained results better. An example of such image for the San Francisco data is shown in Figure 19.

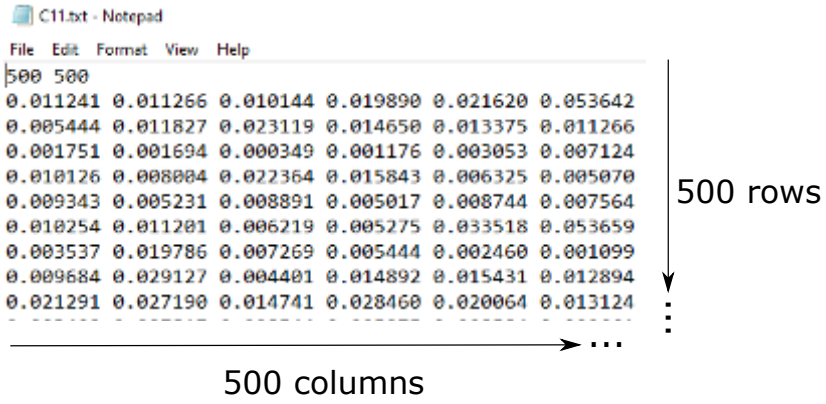


Figure 15: Excerpt from a valid input file (`C11.txt`).

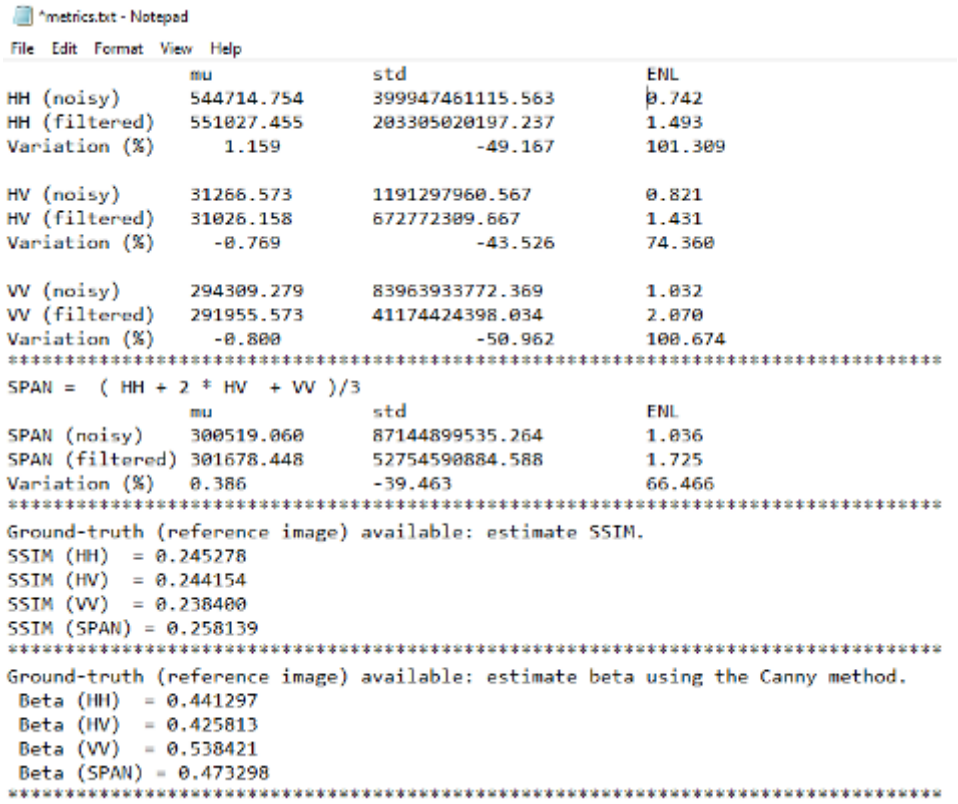


Figure 16: A metrics file (complete and for the case that the user provided a reference image).

Regarding the implementation, a modular programming approach has been followed, that is, the program is separated in several subprograms. Figure 20 shows how the codes are organized. A brief explanation is provided below.

- `Auxiliary_methods.cpp`: it contains some brief methods to perform basic matrix operations and the methods related to get the Pauli representation of the PolSAR data.
- `Edge_detection.cpp`: it contains the two edge detector methods implemented, the Canny and the Laplacian zero-crossing detector.
- `Estimators.cpp`: it contains the estimators used: β and, the SSIM and the non-referenced metrics (preservation of mean value, the variance reduction profile and the ENL).
- `Main.cpp`: it contains the main program.

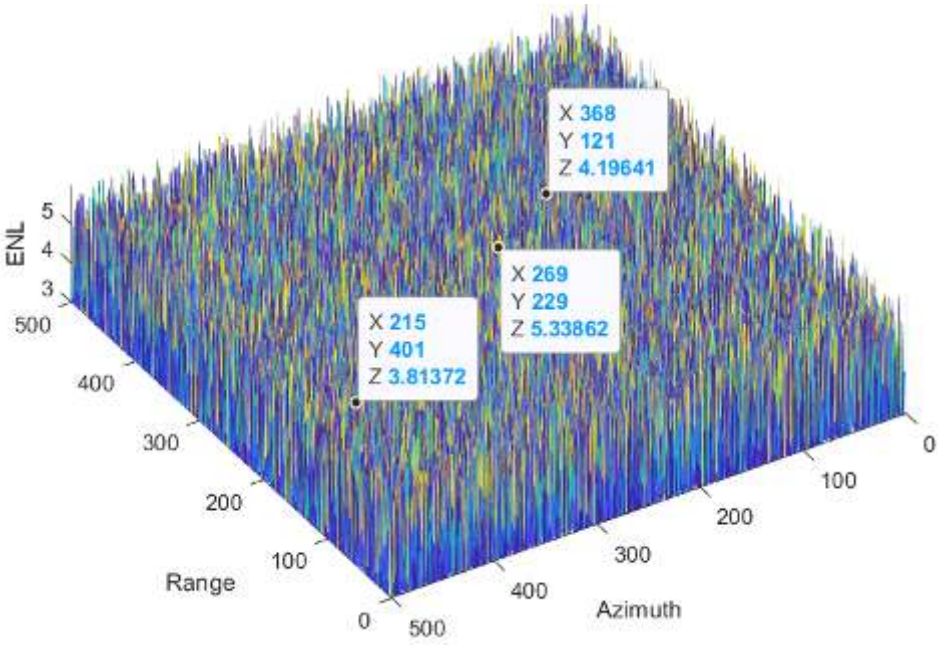


Figure 17: Representation of the ENL estimates for the Flevoland fully PolSAR data.

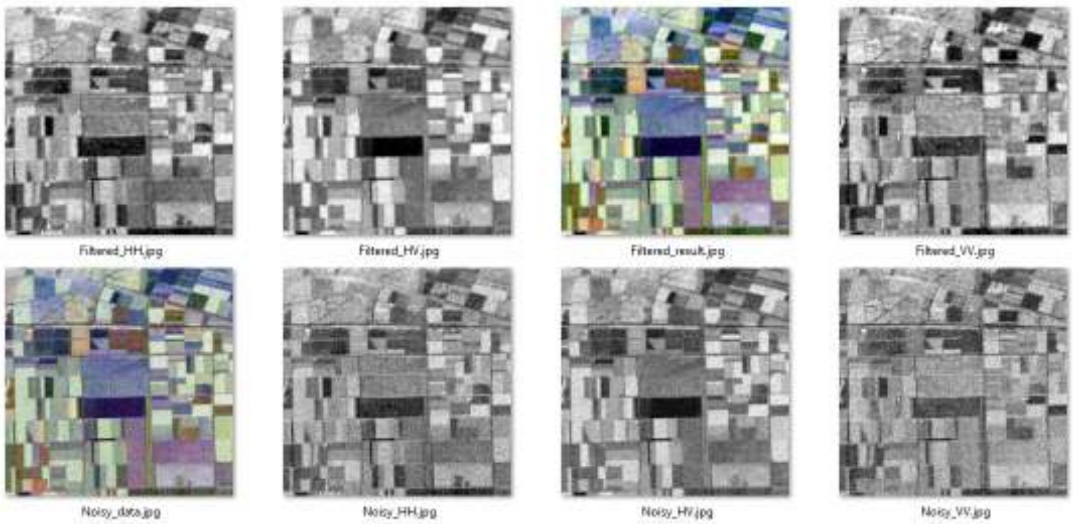


Figure 18: Visualization of the images generated for the Flevoland case.

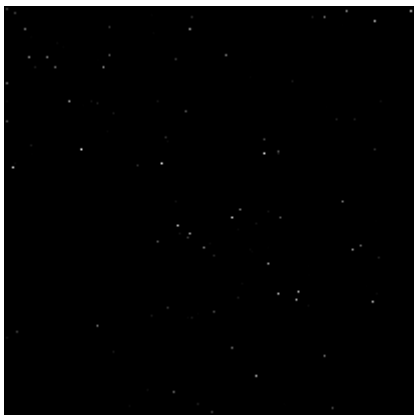


Figure 19: Pixels whose covariance matrix is not positive definite for the San Francisco PolSAR data.

- `PolSAR_Data.cpp`: it contains the methods to load the PolSAR data as well as the non-local PolSAR filter.
- Headers files: the usual C++ headers for each `.cpp` code.

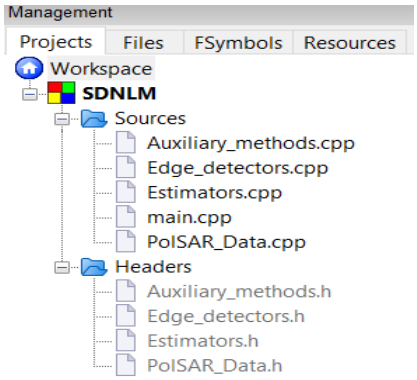


Figure 20: Modular design followed in the implementation of the codes.

As an example to show the benefits of the modular design followed, the call to get the referenced SSIM metric has this form:

```
double ssim_value = SSIM_Estimates (true_data, filtered_data);
```

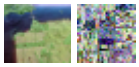
where `true_data` and `filtered_data` are the reference image (the ground truth) and the filtered result respectively. For a user to include a new estimator, just a similar call is required, and the body of the method is inserted in the `Estimators_cpp` subprogram.

Acknowledgement

The authors acknowledge the anonymous reviewers and the editor for their assistance in improving the manuscript's original version and the computational codes.

Image Credits


Synthetic data: provided by the authors.


Actual data: provided by the authors (https://earth.esa.int/web/polsarpro/data-sources/sample-datasets).

References

[1] F. Argenti, A. Lapini, T. Bianchi, and L. Alparone, *A tutorial on speckle reduction in synthetic aperture radar images*, IEEE Geoscience and Remote Sensing Magazine, 1 (2013), pp. 6–35. https://doi.org/10.1109/MGRS.2013.2277512.

[2] A. Buades, B. Coll, and J.M. Morel, *A review of image denoising algorithms, with a new one*, Multiscale Modeling & Simulation, 4 (2005), pp. 490–530. https://doi.org/10.1137/040616024.

[3] ——, *Image denoising methods. A new nonlocal principle*, SIAM Review, 52 (2010), pp. 113–147. https://doi.org/10.1137/090773908.

[4] D. Chan, J. Gambini, and A.C. Frery, *Entropy-based non-local means filter for single-look SAR speckle reduction*, Remote Sensing, 14 (2022), p. 509. https://doi.org/10.3390/rs14030509.

[5] D. Coelho, R. Cintra, A.C. Frery, and V. Dimitrov, *Fast matrix inversion and determinant computation for polarimetric synthetic aperture radar*, Computers & Geosciences, 119 (2018), pp. 109–114. https://doi.org/10.1016/j.cageo.2018.07.002.

[6] C.-A. Deledalle, L. Denis, and F. Tupin, *Iterative weighted maximum likelihood denoising with probabilistic patch-based weights*, IEEE Transactions on Image Processing, 18 (2009), pp. 2661–2672. https://doi.org/10.1109/tip.2009.2029593.

[7] C.-A. Deledalle, V. Duval, and J. Salmon, *Non-local methods with shape-adaptive patches (NLM-SAP)*, Journal of Mathematical Imaging and Vision, 43 (2011), pp. 103–120. https://doi.org/10.1007/s10851-011-0294-y.

[8] X. Deng, C. López-Martínez, and E.M. Varona, *A physical analysis of polarimetric SAR data statistical models*, IEEE Transactions on Geoscience and Remote Sensing, 54 (2016), pp. 3035–3048. https://doi.org/10.1109/TGRS.2015.2510399.

[9] D. Ebert, F. Kenton Musgrave, D. Peachey, K. Perlin, and S. Worley, *Texturing & Modeling: a Procedural Approach*, Morgan Kaufmann, San Francisco, CA, 2003. ISBN 1558608486.

[10] A.C. Frery, *Stochastic contrast measures for SAR data: A survey*, Journal of Radars, 8 (2019), pp. 758–781. https://doi.org/10.12000/JR19108.

[11] A.C. Frery, A.D.C. Nascimento, and R.J. Cintra, *Analytic expressions for stochastic distances between relaxed complex Wishart distributions*, IEEE Transactions on Geoscience and Remote Sensing, 52 (2014), pp. 1213–1226. https://doi.org/10.1109/TGRS.2013.2248737.

[12] G. Gao, *Statistical modeling of SAR images: A survey*, Sensors, 10 (2010), pp. 775–795. https://doi.org/10.3390/s100100775.

[13] L. Gomez-Deniz and A.C. Frery, *The influence of distances in NLM PolSAR filters*, in IEEE International Geoscience & Remote Sensing Symposium (IGARSS), IEEE Press, July 2019, pp. 5109–5112. https://doi.org/10.1109/IGARSS.2019.8900394.

[14] N.R. Goodman, *The distribution of the determinant of a complex Wishart distributed matrix*, Annals of Mathematical Statistics, 34 (1963), pp. 178–180.

[15] Q. Guo, C. Zhang, Y. Zhang, and H. Liu, *An Efficient SVD-Based Method for Image Denoising*, IEEE Transactions on Circuits and Systems for Video Technology, 26 (2016), pp. 868–880. https://doi.org/10.1109/TCSVT.2015.2416631.

[16] M. Hagedorn, P.J. Smith, P.J. Bones, R.P. Millane, and D. Pairman, *A trivariate chi-squared distribution derived from the complex Wishart distribution*, Journal of Multivariate Analysis, 97 (2006), pp. 655–674. https://doi.org/10.1016/j.jmva.2005.05.014.

[17] J.S. Lee, K.W. Hoppel, S.A. Mango, and A.R. Miller, *Intensity and phase statistics of multilook polarimetric and interferometric SAR imagery*, IEEE Transactions on Geoscience and Remote Sensing, 32 (1994), pp. 1017–1028. https://doi.org/10.1109/36.312890.

[18] J.-S. Lee and E. Pottier, *Polarimetric Radar Imaging: From Basics to Applications*, CRC, Boca Raton, 2009. ISBN 9781420054972.

[19] J.S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice Hall Signal Processing Series, Prentice Hall, Englewood Cliffs, 1989. ISBN 978-0-13-935322-2.

[20] G. Liu, H. Zhong, and L. Jiao, *Comparing noisy patches for image denoising: A double noise similarity model*, IEEE Transactions on Image Processing, 24 (2015), pp. 862–872. https://doi.org/10.1109/TIP.2014.2387390.

[21] E. Moschetti, M.G. Palacio, M. Picco, O.H. Bustos, and A.C. Frery, *On the use of Lee's protocol for speckle-reducing techniques*, Latin American Applied Research, 36 (2006), pp. 115–121.

[22] A.D.C. Nascimento, R.J. Cintra, and A.C. Frery, *Hypothesis testing in speckled data with stochastic distances*, IEEE Transactions on Geoscience and Remote Sensing, 48 (2010), pp. 373–385. https://doi.org/10.1109/TGRS.2009.2025498.

[23] W. Ni and X. Gao, *Despeckling of SAR image using generalized guided filter with Bayesian nonlocal means*, IEEE Transactions on Geoscience and Remote Sensing, 54 (2016), pp. 567–579. https://doi.org/10.1109/TGRS.2015.2462120.

[24] L. Pardo, D. Morales, M. Salicrú, and M.L. Menéndez, *Generalized divergence measures: Information matrices, amount of information, asymptotic distribution, and its applications to test statistical hypotheses*, Information Sciences, 84 (1995), pp. 181–198. https://doi.org/10.1016/0020-0255(95)00017-J.

[25] L. Pardo, D. Morales, M. Salicrú, and M.L. Menéndez, *Large sample behavior of entropy measures when parameters are estimated*, Communications in Statistics – Theory and Methods, 26 (1997), pp. 483–501. https://doi.org/10.1080/03610929708831929.

[26] M. Salicrú, M.L. Menéndez, and L. Pardo, *Asymptotic distribution of (h, ϕ)-entropy*, Communications in Statistics - Theory and Methods, 22 (1993), pp. 2015–2031. https://doi.org/10.1080/03610929308831131.

[27] M. Salicrú, D. Morales, and M.L. Menéndez, *On the application of divergence type measures in testing statistical hypothesis*, Journal of Multivariate Analysis, 51 (1994), pp. 372–391. https://doi.org/10.1006/jmva.1994.1068.

[28] C.A.N. Santos, D.L.N. Martins, and N.D.A. Mascarenhas, *Ultrasound Image Despeckling Using Stochastic Distance-Based BM3D*, IEEE Transactions on Image Processing, 26 (2017), pp. 2632–2643. https://doi.org/10.1109/TIP.2017.2685339.

[29] A.K. Seghouane and S.I. Amari, *The AIC criterion and symmetrizing the Kullback-Leibler divergence*, IEEE Transactions on Neural Networks, 18 (2007), pp. 97–106. https://doi.org/10.1109/tnn.2006.882813.

[30] L. Torres, T. Cavalcante, and A.C. Frery, *Speckle reduction using stochastic distances*, in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP), L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, eds., vol. 7441 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2012, pp. 632–639. https://doi.org/10.1007/978-3-642-33275-3_78.

[31] L. Torres, S. J.S. Sant'Anna, C.C. Freitas, and A.C. Frery, *Speckle Reduction in Polarimetric SAR Imagery with Stochastic Distances and Nonlocal Means*, Pattern Recognition, 47 (2014), pp. 141–157. https://doi.org/10.1016/j.patcog.2013.04.001.

[32] S. Vitale, D. Cozzolino, G. Scarpa, L. Verdoliva, and G. Poggi, *Guided Patchwise Nonlocal SAR Despeckling*, IEEE Transactions on Geoscience and Remote Sensing, 57 (2019), pp. 6484–6498. https://doi.org/10.1109/TGRS.2019.2906412.

[33] J. Wu, F. Liu, H. Hao, L. Li, L. Jiao, and X. Zhang, *A nonlocal means for speckle reduction of SAR image with multiscale-fusion-based steerable kernel function*, IEEE Geoscience and Remote Sensing Letters, 13 (2016), pp. 1646–1650. https://doi.org/10.1109/LGRS.2016.2600558.

[34] D.-X. Yue, F. Xu, A.C. Frery, and Y.-Q. Jin, *A generalized Gaussian coherent scatterer model for correlated SAR texture*, IEEE Transactions on Geoscience and Remote Sensing, 58 (2020), pp. 2947–2964. https://doi.org/10.1109/TGRS.2019.2958125.

[35] ——, *Synthetic aperture radar image statistical modeling Part I: Single-pixel statistical models*, IEEE Geoscience and Remote Sensing Magazine, 9 (2021), pp. 82–114. https://doi.org/10.1109/MGRS.2020.3004508.