

Bio-Inspired Optimization Algorithms for Automatic Estimation of Multiple Subspace Dimensions in a Tensor-Wavelet Denoising Algorithm

Abir Zidi¹, Julien Marot², Salah Bourennane², Klaus Spinnler¹

¹Fraunhofer Institute IIS, Produktionsmonitoring, Flugplatzstrasse 75, 90768 Fürth, Germany

²Aix Marseille University, Ecole Centrale Marseille, Institut Fresnel, D. U. de Saint Jérôme Av. escadrille Normandie-Niemen, 13397 Marseille, France

¹abir.zidi@iis.fraunhofer.de; ²julien.marot@fresnel.fr; ³salah.bourennane@fresnel.fr; ⁴klaus.spinnler@iis.fraunhofer.de

Abstract-This paper focuses on the denoising of multidimensional data by a tensor subspace-based method. In a seminal work, multiway Wiener filtering was developed to minimize the mean square error between an expected signal tensor and the estimated tensor. It was then placed in a wavelet packet framework, with the pending issue of a reliable method for the estimation of multiple signal subspace dimensions for multiple coefficients of the wavelet packet transform. For the first time in this paper, we aim at estimating the signal subspace dimensions for all modes of wavelet packet coefficients by minimizing the least squares criterion with the best possible optimization strategy. In the paper, the interest of a genetic algorithm and particle swarm optimization for this purpose are compared. Also, the computational load is reduced as follows: the best subspace dimension values are estimated on subsampled data, and the original data is denoised with values of signal subspace dimension which are scaled by subsampling factors. The results obtained are compared on multispectral images regarding signal-to-noise ratio and perceptual image quality for various noise levels: the proposed method outperforms the existing ones for various multispectral images, containing different numbers of bands. An application to plant leaf fluorescence image denoising is included.

Keywords- Image Denoising; Multispectral Imaging; Optimization

I. INTRODUCTION

Multispectral images are now used, in life science for instance (see [1] and references inside), but also in remote sensing applications. In this context, devoted sensors have been developed, such as ROSIS sensor (Reflective Optics System Imaging Spectrometer) [2], or AVIRIS sensor (Airborne Visible/Infrared Imaging Spectrometer) [3].

The multispectral image can be obtained by selecting some important bands from the hyperspectral images obtained by these airborne sensors. Most of the multispectral aerial images are impaired by noise [4, 5] from solar radiation, or atmospheric scattering [6] for instance. Multispectral images acquired in the life science context may also be noisy in the case where the imaged phenomenon, such as fluorescence [1], yields a low luminous intensity, or when the light is scattered, such as in some medical issues [7]. Because of the presence of noise in hyperspectral or multispectral images, there exists a growing interest in developing algorithms to denoise such images.

Relation with previous work in the field: A seminal work consisted in adapting Wiener filtering in a tensor framework, yielding the Multiway Wiener Filtering (MWF) [8], a subspace-based method requiring the estimation of the dimension of the signal subspace, also called rank, along with each mode. This estimation is usually performed with the statistical Akaike information criterion (AIC) [9], working best with a very high number of signal realizations. Recently, the MWPT-MWF (Multidimensional Wavelet Packet Transform-Multiway Wiener Filtering) method has been proposed [4, 10, 11], yielding good results regarding signal to noise ratio (SNR) improvement and classification accuracy. Wavelet packet transform (WPT) is applied to a signal over a certain number of levels, similarly to discrete wavelet transform (DWT). However, when the DWT is applied, only the approximation coefficient is decomposed from one level to the next. When WPT is applied, each coefficient, either approximation or detail, is decomposed itself into an approximation and a detail coefficient from one level to the next.

In this work, we use Multidimensional Wavelet Packet Transform (MWPT), the extension of WPT to multidimensional arrays such as multispectral images (see [4] and references inside). When MWPT is applied, each coefficient is a multidimensional array.

The drawback of MWPT-MWF is that, due to a large number of coefficients which are computed, a large number of signal subspace dimension values must be estimated to ensure accurate denoising results. In [4], a study about the accurate depth of the wavelet packet decomposition has been performed, but the signal subspace dimensions are still estimated with AIC. In [12], the minimization of the least squares criterion is proposed to estimate the rank values. This criterion is a squared difference

between the expected image and a reference image obtained by a rough denoising of the raw image to process. The relevance of this criterion and its optimality are not proven. Also, the problem of the high computational load of the proposed methods was still pending, as well as its dependence on the reference image involved in the criterion. As for the applications considered up to now in related papers, to the best of our knowledge, the considered values of SNR in the processed data are rather elevated [4, 13]. Indeed, the last generation of airborne sensors provide images of high quality.

Goal and contributions: In this context, where a criterion should be minimized while considering the computational load, this paper offers the following contributions: Firstly, we compare several optimization methods such as genetic algorithm (GA) using a Lagrangian algorithm [14, 15] or particle swarm optimization (PSO) [16], but also Nelder-Mead [17], considering whether they are adequate to estimate the signal subspace dimensions in MWF. Secondly, we propose an unsupervised method, named MWPT-MWF-RE (Multidimensional Wavelet Packet Transform-Multiway Wiener Filtering with Rank Estimation). For the first time in this context, we aim at reducing the computational load, by estimating the required rank values on subsampled data. Thirdly, the adequate number of decomposition levels for wavelet packet decomposition is estimated automatically, on subsampled data as well. Also, the algorithm becomes free from the dependence on the reference image thanks to an iterative process.

As the proposed methods perform best for rather low SNR values regarding SNR improvement, we propose, as a last contribution of this paper, an application to the denoising of fluorescence images acquired from plant leaves. In this context, the relevant light emitted by the leaves is of very low intensity, hence the low SNR values obtained in some conditions of acquisition.

Outline: Section II. sets the problem of the estimation of multiple values of signal subspace dimensions for MWF in a wavelet framework: we propose a criterion to minimize, highlight the need for a global optimization method for this purpose, and propose a literature review on bio-inspired stochastic optimization methods. In Section III., we show how to adapt a genetic algorithm or particle swarm optimization to minimize the proposed criterion. In Section IV., this algorithm is implemented as follows: the optimal values for signal subspace dimension are first estimated on a subsampled version of the data, and scaled before their use for multiway Wiener filtering of multidimensional wavelet packet coefficients. We propose an unsupervised MWF in MWPT: we show the interest of the automatic estimation of multiple ranks, and how to refine the reference image required in the proposed criterion. In Section V., we show the denoising results obtained with the proposed optimization algorithms on various types of multidimensional images: an RGB image, multispectral images obtained from the ROSIS sensor, and a multispectral acquisition of a plant leaf fluorescence phenomenon. These results are compared results with those obtained with other subspace-based methods such as the truncation of higher-order singular value decomposition (HOSVD), MWF, but also Perona-Malik [18] which is based on a diffusion process.

Notations: The following notations are used in the rest of the paper: scalars are denoted by italic lowercase roman, like a ; vectors by boldface lowercase roman, like \mathbf{a} ; matrices by boldface uppercase roman, like \mathbf{A} ; tensors by uppercase, like \mathbf{A} ; and the n -mode product, denoted by $\times_1, \times_2, \times_3$ for modes 1, 2, and 3 respectively, is defined as the product between a tensor \mathbf{A} and a matrix \mathbf{B} in mode n .

II. STATE-OF-THE-ART AND PROBLEM SETTING

The accurate estimation of signal subspace ranks for the purpose of multidimensional data is still pending and has become a required step since multiway Wiener filtering has been inserted in a wavelet framework. To solve this issue, we propose a criterion to be minimized with a global optimization method.

A. Rank Estimation Issue in a Multidimensional Context

A noisy multidimensional signal will be considered, also called tensor: a signal \mathbf{X} impaired by a multidimensional additive, zero-mean white Gaussian noise \mathbf{N} [19]. For instance, the additive case holds for hyperspectral images [9, 20]. As concerns the white noise assumption, it is also generally adopted for multidimensional images [19], and permits to focus on the main issue of this paper. In the case where the noise is not white, a prewhitening process could be applied as proposed in [2]. Thus, this tensor can be a model for an HSI, expressed as: $\mathbf{R} = \mathbf{X} + \mathbf{N}$. Tensors \mathbf{R} , \mathbf{X} , and \mathbf{N} are of size $I_1 \times I_2 \times I_3$. For each spectral band indexed by $i = 1, \dots, I_3$, the noise $\mathbf{N}(:, :, i)$ is assumed additive, zero-mean, white and Gaussian. The objective is to denoise the tensor \mathbf{R} with a subspace-based method. Subspace-based methods have been shown to exhibit good denoising results when applied to data with main salient orientations in the image [21]. They provide an estimated signal tensor which, generally in the literature and the remainder of this paper, is denoted by $\hat{\mathbf{X}}$. This estimate depends on the so-called signal subspace dimensions' or signal subspace ranks' $\{K_1, K_2, K_3\}$ which must be estimated.

B. Multiway Wiener Filtering in the Wavelet Framework

In [1], MWF has been inserted into a wavelet framework to denoise images while preserving details. In tensor form, 2-dimensional wavelet packet transform (WPT) is applied for each mode to get the multidimensional wavelet packet transform (MWPT):

$$C_l^R = R \times_1 W_1 \times_2 W_2 \times_3 W_3 \quad (1)$$

In Eq. (1), C_l^R is the wavelet packet coefficient tensor for levels in $l = [l_1, l_2, l_3]^T$; for each value of n between 1 and 3, $W_n \in IR^{l_n \times l_n}$ indicate the WPT matrices, and \times_n denotes the n -mode product [4].

To select the ‘frequency’ components out of the wavelet packet coefficient tensor, a ‘frequency’ index is defined for the mode 1, 2 or 3. A set of indices forms the index vector $\mathbf{m} = [m_1, m_2, m_3]^T$. For n equal to 1, 2 or 3, the index value m_n is such that $0 \leq m_n \leq 2^{l_n} - 1$.

The coefficient subtensor extracted from C_l^R and containing the frequency components for a given index vector \mathbf{m} is denoted by $C_{l,m}^R$. It is worth noticing that the size of each coefficient $C_{l,m}^R$ is $\frac{l_n}{2^{m_n}}$ along mode n , whatever the index vector \mathbf{m} [4].

The principles of multidimensional wavelet packet transform-multiway Wiener filtering (MWPT-MWF) proposed in [4] are to apply MWF to each coefficient subtensor $C_{l,m}^R$. Following the same notation as for the noised tensor R , let $C_{l,m}^X$ and $\hat{C}_{l,m}^X$ be the coefficient subtensors for the expected tensor X and its estimate \hat{X} respectively. From the Parseval theorem, minimizing the MSE between X and its estimate \hat{X} is equivalent to minimizing the MSE between $C_{l,m}^X$ and $\hat{C}_{l,m}^X$ for each \mathbf{m} :

$$\|X - \hat{X}\|^2 = \|C_l^X - \hat{C}_l^X\|^2 = \sum_{\mathbf{m}} \|C_{l,m}^X - \hat{C}_{l,m}^X\|^2 \quad (2)$$

To minimize the expression $\|C_{l,m}^X - \hat{C}_{l,m}^X\|^2$ for all instances of vector \mathbf{m} , multiway Wiener filtering is applied. This requires, for each vector \mathbf{m} , the knowledge of three rank values.

C. Proposed Criterion for Optimal Rank Selection

In the literature, the Akaike Information Criterion (AIC) [4] is commonly used to estimate the subspace ranks. AIC estimates the number of sources correctly in an array processing problem [22]. In an array processing context, there are many realizations of the same random signal, and the data are rarely noisy, so AIC exhibits a good behavior. Usually, in the frame of HSI processing, through a stationarity hypothesis, a covariance matrix is computed from the column vectors of the unfolded matrix obtained from the HSI, which are considered as realizations of the same random signal. AIC is applied to the eigenvalues of the covariance matrix obtained for each mode of the HSI [9]. However, it has been shown empirically that there is no clear domination of a subset of eigenvalues with high magnitude with respect to the others [9], and that AIC tends to overestimate the rank values when the data are noisy.

In [12], the minimization of the least squares criterion is proposed to estimate the rank values, but its relevance and optimality are not proven. We look for a method which is robust to noise and takes into account the assumptions about our noise model. Maximum likelihood restoration methods have been developed during the last decades for blurred, Poisson noise or additive Gaussian noise models [23]. In the case of Gaussian noise, they yield to the minimization of the least squares criterion [24]. Among other methods [24], mean square error minimization yielded the Wiener filter [25], and MWF [19], which is the core of the methods presented in this paper. For these reasons, and because our model for noise is white Gaussian, we propose to minimize a least square error (LSE) criterion to estimate the signal subspace dimensions:

$$J_m(K_1, K_2, K_3) = \|C_{l,m}^{X_1} - \hat{C}_{l,m}^X(K_1, K_2, K_3)\|^2 \quad (3)$$

where $\|\cdot\|$ represents the Frobenius norm, X_1 is a gross estimate of the expected tensor X , and $\hat{C}_{l,m}^X$ is a coefficient subtensor of the final estimate \hat{X} . $\hat{C}_{l,m}^X$ depends on the expected rank values (K_1, K_2, K_3) . It is obtained with MWF, possibly including fixed point algorithm [26].

The criterion $J_m(K_1, K_2, K_3)$ is a nonlinear function of the parameters K_1, K_2, K_3 , hence the need for an adequate optimization method, which must be global. In [12], a single optimization method, particle swarm optimization, is proposed. To the best of our knowledge, no thorough investigation was performed on the comparative performance of optimization methods to estimate the signal subspace ranks in a wavelet framework.

D. Short Literature Review on Global Optimization Methods

The considered optimization problem is highly non-linear, and we have no insurance that some constraints on the minimized function are respected. Therefore, we left aside the deterministic optimization methods such as Dividing Rectangles (DIRECT) [27] which assumes that the minimized function is a k -Lipschitzian one. Moreover, as the data considered in this work are impaired with random noise, we focused on stochastic optimization methods. These methods introduce randomness in the initialization and iterative process of the optimization algorithm. Contrary to well-known deterministic methods such as Gradient descent which may converge to a local minimum, this randomness prevents the stochastic optimization method from focusing on a local optimum, but also turns it less sensitive to modeling errors. Further in this paper, we provide some details on two bio-inspired optimization methods: swarm optimization [16] and genetic algorithms [28]. First, we present Nelder-Mead [17]. The Nelder-Mead Simplex method [17] is a global optimization method, which is meant to minimize a scalar-valued nonlinear function of several real variables, without any derivative information. It is known to yield a rapid decrease in

cost function values [17]. It has been shown that, in dimension two or more, the Nelder-Mead method may converge to a non-critical point of the minimized function [29]. In their seminal work concerning particle swarm optimization, Kennedy and Eberhart [16, 30] got inspired by [31], where the term ‘particle swarm’ was chosen to define the members of a population or test set. In their paradigm, the population members are mass-less and volume-less. Their evolution is described through position, speed, and acceleration parameters [16]. An often cited, now well-known reference [28] introduces genetic algorithms in the context of evolutionary computation which implies the evolution of a population of candidates which is inspired by Darwin’s natural selection theory. Another largely cited reference presents basics about genetics, the hierarchical genetic algorithm, and applications to H^∞ control, neural network, and speech recognition [32].

For applications which concern specifically image processing, genetic algorithms are combined with mathematical morphology methods [33], watermarking [34], medical imaging [35], or 3D reconstruction [36].

As GA requires a large number of function evaluations for convergence, and because PSO may prematurely converge to a local minimum, hybrid algorithms were proposed which combine the advantages of both types of methods: genetic algorithms were mixed with particle swarm optimization [37-41], or an ant colony algorithm [42].

Recently, such a hybrid algorithm was proposed [43], where the evolutionary natures and social interactions of both algorithms are combined in the frame of a multibiometric system.

Nelder-Mead can be a valuable comparative method. However, three parameters should be estimated in the considered optimization problem, and Nelder-Mead method could be stacked in a local minimum of the criterion in Eq. (3). Hence, and as a balance to subsection D., we will then focus on two types of stochastic bio-inspired optimization methods: genetic algorithm (GA) and particle swarm optimization (PSO). Both types of methods provide the global minimum of a scalar function of several variables and are gradient-free.

III. GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION FOR TENSOR RANK ESTIMATION

GA and PSO are population-based iterative algorithms. They start with an initial random set of Q rank triplets called population. In the following, this population is denoted by $y_q^{K_1, K_2, K_3}$, $q = 1, \dots, Q$. Each vector $y_q = [K_1, K_2, K_3]^T$ is modified throughout the iterations. Vector y_q is called a chromosome while dealing with a genetic algorithm, and a particle while dealing with swarm optimization.

A. Principles of Genetic Algorithms

The genetic algorithm is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution. It is based on ‘survival of the fittest’ theory of Charles Darwin [44]. It was introduced by Holland in 1975 [45].

This algorithm is interesting because it’s very robust in nature and capable of optimizing complex results, including when a great number of unknown values are expected.

Each individual $y_q^{K_1, K_2, K_3}$, $q = 1, \dots, Q$ in the population is called a chromosome, representing a solution to problem at hand. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes $y_q^{K_1, K_2, K_3}$ are evaluated using the fitness function $J_m(y_q^{K_1, K_2, K_3})$, directly adapted from Eq. (3):

$$J_m(y_q^{K_1, K_2, K_3}) = \|C_{l,m}^{X_1} - \hat{C}_{l,m}^X(y_q^{K_1, K_2, K_3})\|^2 \quad (4)$$

The genetic algorithm uses three main types of rules at each step to create the next generation from the current population [46]:

- Selection rules select the individuals, called parents, that are best adapted to their environment and contribute to the population at the next generation.
- Crossover is a genetic operator that recombines two chromosomes (a pair of individuals) to produce a new chromosome (called an offspring).

The new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution according to a user-definable crossover probability.

- Mutation promotes diversity in population characteristics, to prevent the algorithm from being trapped in local minima [43].

The five main steps of a GA algorithm [28] are described in Algorithm 1.

Algorithm 1 Pseudo-code: Genetic Algorithm for multiple rank estimation

Inputs: fitness function of Eq. (4), tensor gross estimate X_1 , small factor ϵ set by the user, to stop the algorithm.

1. Set iteration number $iter = 1$, and maximum number of iterations $maxiter$.
2. Create an initial population composed of Q random chromosomes with all required rank values $y_q^{K_1, K_2, K_3}(iter)$, $q = 1, \dots, Q$. This initial population takes the form of a matrix with Q rows and 3 columns.
3. Evaluate fitness function value of Eq. (3) for each chromosome $y_q^{K_1, K_2, K_3}(iter)$ in the population
4. Generate the new population:
 - (a) Selection: Choose new parents through fitness function value.
 - (b) Crossover: Pair probabilistically the parents to create the offsprings with crossover fraction X_f .
 - (c) Mutation: Modify slightly some components, chosen randomly, of each offspring. We now afford the population $y_q^{K_1, K_2, K_3}(iter + 1)$, $q = 1, \dots, Q'$.
5. Replace the current population with the new one, obtained at step 4.
6. Increase $iter$; if $iter < maxiter$, or $\|y_q^{K_1, K_2, K_3}(iter) - y_q^{K_1, K_2, K_3}(iter - 1)\| > \epsilon$ go to step 3.

Output: estimated rank values $\widehat{K}_1, \widehat{K}_2, \widehat{K}_3$.

Here are some details for GA algorithm:

At each iteration $iter$, $y_q^{K_1, K_2, K_3}(iter)$ is the current chromosome q .

At step 2, the initial population is stored in a $Q \times 3$ matrix. Each row of the matrix is 1×3 vector with three random values. The n^{th} component of this vector is a random value between 1 and $\frac{I_n}{2^{ln}}$. Therefore, the q^{th} row of the matrix is $y_q^{K_1, K_2, K_3}(1)$.

In the following, we exemplify the step 4 of Algorithm 1, with two chromosomes with index q and q' :

At step 4, after selection, we assume that we afford, among the selected chromosomes, $y_q^{K_1, K_2, K_3}(iter) = [K_1, K_2, K_3]$ and $y_{q'}^{K_1, K_2, K_3}(iter) = [K_1', K_2', K_3']$.

At the iteration $iter + 1$, crossover provides the new offsprings. Whatever n , either K_n or K_n' is chosen, randomly, as the n^{th} component of $y_q^{K_1, K_2, K_3}(iter + 1)$ and $y_{q'}^{K_1, K_2, K_3}(iter + 1)$. For instance: $y_q^{K_1, K_2, K_3}(iter + 1) = [K_1, K_2', K_3']$ and $y_{q'}^{K_1, K_2, K_3}(iter + 1) = [K_1', K_2, K_3']$.

Then after mutation, these offsprings are slightly modified: $y_q^{K_1, K_2, K_3}(iter + 1) = [K_1 + \kappa_1, K_2' + \kappa_2, K_3' + \kappa_3]$ and $y_{q'}^{K_1, K_2, K_3}(iter + 1) = [K_1' + \kappa_1', K_2 + \kappa_2', K_3 + \kappa_3']$, where, whatever n , κ_n and κ_n' are random values which are small compared K_n and K_n' .

For the practical implementation of Algorithm 1, we have selected a globally convergent Lagrangian algorithm which permits to define lower and upper bounds on the estimated values $\widehat{K}_1, \widehat{K}_2, \widehat{K}_3$ [15]. Our motivation is the following: in the fitness function of Eq. (4), the values of K_1, K_2, K_3 cannot be less than 1, and greater than the size of the wavelet packet coefficient $\widehat{C}_{l,m}^{X_1}$ along each mode n : $0 < K_n \leq \frac{I_n}{2^{ln}}$, where n is equal to either 1, 2, or 3.

B. Principles of Particle Swarm Optimization Algorithm

The basic PSO algorithm [16] is inspired by the behavior of fish or bird swarms. The basic principles of PSO algorithm are to simulate the communication between animals of a swarm, which aim at locating food for instance. PSO updates the behavior of such individuals of the swarm, called particles, through time. As shown in Algorithm 2, it is implemented as an iterative algorithm which, for the current iteration number $iter$, computes two characteristics of the particles: velocity, and position.

Algorithm 2 Pseudo-code: Particle Swarm Optimization for multiple rank estimation

Inputs: fitness function of Eq. (4), tensor gross estimate X_1 , small factor ϵ set by the user, to stop the algorithm.

1. Set iteration number $iter = 1$, create an initial population composed of Q random particles with all required rank values $y_q^{K_1, K_2, K_3}(iter)$, $q = 1, \dots, Q$. This initial population takes the form of a matrix with Q rows and 3 columns.
2. Evaluate fitness function value of each particle $y_q^{K_1, K_2, K_3}(iter)$, $q = 1, \dots, Q$.
3. Update the local best particles $p_q^{K_1, K_2, K_3}(q = 1, \dots, Q)$, and the global best particle G^{K_1, K_2, K_3} ,

4. Repeat steps for each particle $q, q = 1, \dots, Q$:
 - (a) Compute velocity $V_q^{K_1, K_2, K_3}(iter + 1)$
 - (b) Compute position $y_q^{K_1, K_2, K_3}(iter + 1)$
5. Exchange current population with the new one, obtained at step 4.
6. If $iter < maxiter$, or $||y_q^{K_1, K_2, K_3}(iter) - y_q^{K_1, K_2, K_3}(iter - 1)|| > \epsilon$ increase it , and go to step 2.

Output: estimated rank values $\widehat{K}_1, \widehat{K}_2, \widehat{K}_3$

Here are some details for PSO algorithm:

At each iteration $iter$, $y_q^{K_1, K_2, K_3}(iter)$ is the current position of particle q , and $V_q^{K_1, K_2, K_3}(iter)$ is its velocity.

At step 1, the initial population is stored in a $Q \times 3$ matrix. Each row of the matrix is a 1×3 vector with three random values. The n^{th} coefficient of this vector is a random values between 1 and $\frac{1}{2^n}$. Therefore, at iteration 1, the q^{th} row of the matrix is $y_q^{K_1, K_2, K_3}(1)$.

In the following, we exemplify the step 4. of Algorithm 2, with one particle with index q . This particle is expressed as $y_q^{K_1, K_2, K_3}(iter) = [K_1, K_2, K_3]$. In Eqs. (5) and (6), we show how to get an updated particle, at iteration $iter + 1$.

At step 4(a), the velocity is computed as follows:

$$V_q^{K_1, K_2, K_3}(iter + 1) = W V_q^{K_1, K_2, K_3}(iter) + \gamma_{1q} r_{1q} (p_q^{K_1, K_2, K_3} - y_q^{K_1, K_2, K_3}(iter)) \quad (5)$$

$$\dots + \gamma_{2q} r_{2q} (G^{K_1, K_2, K_3} - y_q^{K_1, K_2, K_3}(iter))$$

At step 4(b), the position at $iter + 1$ is computed as the summation of the position at iteration $iter$ and the velocity at iteration $iter + 1$.

$$y_q^{K_1, K_2, K_3}(iter + 1) = y_q^{K_1, K_2, K_3}(iter) + V_q^{K_1, K_2, K_3}(iter + 1) \quad (6)$$

In Eqs. (5) and (6), the arithmetic operations are computed component-wise. For a given particle q , the velocity is influenced by two contributions: the cognitive one and the social one. The cognitive contribution is due to the personal best location of the particle over all iterations. It is denoted by $p_q^{K_1, K_2, K_3}$ in Eq. (5). Social contribution represents global best location over all particles in the swarm and all iterations. It is denoted by G^{K_1, K_2, K_3} in Eq. (5).

In Eq. (5), W is the inertia weight, γ_{1q} and γ_{2q} are the acceleration constants encouraging a local and a global search respectively. The probabilistic aspect of the transition between two iterations of Algorithm 2 is due to r_{1q} and r_{2q} , which, for each particle q and each iteration $iter$, are two numbers generated randomly between 0 and 1. A large inertia weight (W) facilitates a global search while a small inertia weight facilitates a local search. We look forward to encourage a global search for the first iterations, and a local search for the last iterations. Therefore, we fix an initial value W_{init} and a final value W_{final} for the weighting coefficient. At the iteration $iter$, the weighting coefficient is computed as: $W = W_{init} - \frac{(W_{init} - W_{final}) * iter}{maxiter}$, where $maxiter$ is the final iteration number. When this last iteration number is attained, or when the Frobenius norm of the difference between estimated vectors at iteration $iter$ and $iter - 1$ is less than a fixed threshold ϵ (see step 6 of Algorithm 2), the position vector G^{K_1, K_2, K_3} is a row vector containing 3 components which are the final estimated values $\widehat{K}_1, \widehat{K}_2, \widehat{K}_3$, of the signal subspace ranks.

C. Comparative Discussion: GA and PSO

From Algorithms 1 and 2, we can learn that GA and PSO share many common points: they search a population of points in parallel, not a single point; they do not require derivative information or other auxiliary knowledge, only the objective function. They use probabilistic transition rules between one iteration to the next, not deterministic ones. This probabilistic aspect is due to the probabilistic crossover and mutation at steps 4(b) and 4(c) in Algorithm 1 (GA). The role of mutation is often seen as providing a guarantee that the probability of searching any given triplet of rank values will never be zero. The crossover routines recombine pairs of individuals with given probability to produce offspring. In Algorithm 2 (PSO), the probabilistic aspect is due to random parameters r_{1q} and r_{2q} (see Eq. (5) and step 4(b) of Algorithm 2). The role of these parameters is to enable the scan of the whole research space and to avoid the algorithm to be trapped in a local minimum.

Different terms are used for similar features in both types of algorithms: both methods start with a population of randomly generated candidate solutions, called individuals or phenotypes in GA, and particles in PSO. The population is evolved toward better solutions, regarding a fitness function. Each candidate solution has a set of properties, called chromosomes or genotype in GA, position, and velocity in particle swarm optimization.

Both update the population and search for the optimum with random techniques. Both systems do not guarantee success. GA and PSO are different mainly in the way the information is shared between candidate solutions. In a nutshell, advantages and drawbacks of both methods are as follows: In GA, candidate solutions with much diversity are created. Every chromosome shares information with at least another chromosome, or mutates. The reproductive success varies with fitness, but the best solution at a given iteration is not preserved. Hence, a large number of iterations are generally required for convergence [43].

In PSO, particles do not undergo crossover and mutation: they update themselves with the internal velocity, which depends on local $p_q^{K_1, K_2, K_3}$ and global G^{K_1, K_2, K_3} best candidates selected at the previous iteration (see Eqs. (5) and (6)). There exists a single particle G^{K_1, K_2, K_3} which has an influence on the behavior of the other candidate solutions. Hence, PSO includes a ‘memory’ feature: over all iterations, the personal best of each particle and the overall best are taken into account to compute velocity. So, the evolution only looks for the best solution at a given iteration. In PSO, there is no selection operator, and the higher the number of particles, the higher the computational load. However, compared with GA, and even if the number of particles remains the same throughout the evolution, the particles tend to converge faster to the final solution.

In the considered problem, we look for only three parameters which are the signal subspace ranks. Hence, and as a balance to section III., we infer that PSO should be faster than GA or any hybrid algorithm. In the next section, we insert the optimization algorithm in a tensor-wavelet framework.

IV. UNSUPERVISED MULTIWAY WIENER FILTERING IN WAVELET FRAMEWORK

In this section, we show how to obtain an unsupervised MWF-MWPT algorithm. We estimate the multiple signal subspace dimensions automatically. We also propose a method to set the number of wavelet packet decomposition levels for each mode, and we refine the reference tensor with a recursive algorithm.

A. Automatic Estimation of Multiple Ranks in Wavelet Framework

It is necessary to adapt a global optimization method for multiple rank estimations when MWF is inserted in the wavelet framework [4]. Indeed, even if we restrict the study to third-order tensors such as color, multispectral or hyperspectral images, the number of required rank values may be so elevated that they cannot be fixed manually: One triplet of rank values must be estimated for each wavelet packet coefficient. There exists 2^{ln} coefficients for mode n , $n = 1, 2, 3$. Each wavelet packet coefficient such as $C_{l,m}^R$ is a tensor of order 3 [4], so 3 rank values must be estimated for each wavelet packet coefficient. When the decomposition is performed, the number of coefficients is equal to $\prod_{n=1}^3 2^{ln}$. In total, the number of rank values to be estimated in the whole algorithm is given by $3 \prod_{n=1}^3 2^{ln}$. This number is possibly elevated. Therefore, it is of great interest to estimate the rank values automatically.

We wish to minimize all terms of the summation in Eq. (2), knowing that the noise-free tensor X is not available. For this we propose Algorithm 3, *multidimensional wavelet packet transform and multiway Wiener filtering with rank estimation* (MWPTMWF-RE).

Algorithm 3 MWPT-MWF-RE

Input: Noisy tensor R , gross estimate X_1 of R obtained by MWF with fixed rank values.

1. Compute the wavelet packet decomposition of the noisy tensor R and of the gross estimate X_1 :

$$C_l^R = R \times_1 \mathbf{W}_1 \times_2 \mathbf{W}_2 \times_3 \mathbf{W}_3, C_l^{X_1} = X_1 \times_1 \mathbf{W}_1 \times_2 \mathbf{W}_2 \times_3 \mathbf{W}_3$$
2. Extract the wavelet packet coefficients [4]: $C_{l,m}^R = C_l^R \times_1 \mathbf{E}_{m_1} \times_2 \mathbf{E}_{m_2} \times_3 \mathbf{E}_{m_3}$, and $C_{l,m}^{X_1} = C_l^{X_1} \times_1 \mathbf{E}_{m_1} \times_2 \mathbf{E}_{m_2} \times_3 \mathbf{E}_{m_3}$
3. Perform rank estimation and denoising for each wavelet packet coefficients $C_{l,m}^R$:

(a) Subsample $C_{l,m}^R$ and $C_{l,m}^{X_1}$ by factors S_1 , S_2 , and S_3 along the first, second and third mode respectively to get a smaller version $C_{l,m}^{RS}$ and $C_{l,m}^{X_1S}$.

(b) Estimate with a global optimization method the optimal rank values $\widehat{K}_1, \widehat{K}_2, \widehat{K}_3$ in terms of the criterion:

$$J_m(K_1, K_2, K_3) = \|C_{l,m}^{X_1S} - \widehat{C}_{l,m}^{XS}(K_1, K_2, K_3)\|^2$$

In the equation above, $\widehat{C}_{l,m}^{XS}$ is obtained from $C_{l,m}^{RS}$ with MWF: $\widehat{C}_{l,m}^{XS} = C_{l,m}^{RS} \times_1 \mathbf{H}_{1,m} \times_2 \mathbf{H}_{2,m} \times_3 \mathbf{H}_{3,m}$, where $\mathbf{H}_{1,m}, \mathbf{H}_{2,m}$, and $\mathbf{H}_{3,m}$ depend on K_1, K_2 , and K_3 respectively.

(c) Multiply the rank values obtained at step (b) by S_1 , S_2 , and S_3 respectively: $\widehat{K}_1 = S_1 * \widehat{K}_1, \widehat{K}_2 = S_2 * \widehat{K}_2$, and $\widehat{K}_3 = S_3 * \widehat{K}_3$.

(d) Apply MWF to each coefficient subtensor $C_{l,m}^R$, with the optimal rank values:

$$\widehat{C}_{l,m}^X = C_{l,m}^R \times_1 \mathbf{H}_{1,m} \times_2 \mathbf{H}_{2,m} \times_3 \mathbf{H}_{3,m}, \text{ where } \mathbf{H}_{1,m}, \mathbf{H}_{2,m}, \text{ and } \mathbf{H}_{3,m} \text{ depend on } \widehat{K}_1, \widehat{K}_2, \text{ and } \widehat{K}_3 \text{ respectively}$$

4. Concatenate all coefficients $\hat{C}_{l,m}^X$ to afford the wavelet packet decomposition \hat{C}_l^X as a whole [4].
5. Retrieve the final estimated tensor by inverse wavelet packet transform: $\hat{X} = \hat{C}_l^X \times_1 \mathbf{W}_1^T \times_2 \mathbf{W}_2^T \times_3 \mathbf{W}_3^T$.

Output: Denoised tensor \hat{X} .

In Algorithm 3, $\mathbf{H}_{1,m}, \mathbf{H}_{2,m}, \mathbf{H}_{3,m}$ denote the n -mode filters of MWF, which depend on rank values K_1, K_2, K_3 [19]: $C_{l,m}^R$ denote the wavelet packet coefficients of R. Note that, for computational load purposes, different versions of MWF may be used in the algorithm. Here are some details about the key steps of Algorithm 3:

At step 1, we compute wavelet packet decomposition of the processed tensor R, to get C_l^R , and the wavelet decomposition of the gross estimate X 1 to get $C_l^{X_1}$.

At step 2, we extract all the coefficients out of the whole decomposition: we get coefficients $C_{l,m}^R$ and $C_{l,m}^{X_1}$. This process is described in detail in [4].

At step 3, we firstly estimate the best rank values, and secondly use these rank values to denoise the coefficients with MWF.

To reduce the computational load of step 3(b), the estimation of the best rank values is performed on subsampled versions of $C_{l,m}^R$ and $C_{l,m}^{X_1}$. The subsampling is made at step 3(a). In step 3(b), we seek for the best rank values, without expecting the best estimates of the singular vectors used in MWF. That is why, still to reduce the computational load, a fast version using fixed-point algorithm and inverse power method [26] may be used if significant data are processed with low SNR values. At step 3(c) we multiply the rank values which were obtained by the subsampling factors. Then, we perform denoising at step 3(d). At step 3(d), the singular vectors used in the Wiener filters along each mode are computed through singular value decomposition, and are then optimal in the least squares sense [47].

At step 4 a simple concatenation of the wavelet packet coefficients is performed. The whole set of coefficients is expressed in a synthetic way as \hat{C}_l^X , which is by definition the wavelet packet decomposition of the denoised tensor.

At step 5 the denoised tensor is provided by inverse wavelet packet transform.

With a global optimization method, Algorithm 3 is supposed to converge asymptotically towards the best set of rank values, and therefore the best possible estimate \hat{X} at step 5 and as the final output of the algorithm. In practice, the total number of iterations, that is, the parameter *maxiter*, is fixed automatically: whatever the optimization algorithm which is considered, the algorithm stops if the maximum *a priori* fixed number of iteration is attained or if the estimated values do not vary from one iteration to the next.

B. Estimation of the Number of Decomposition Levels

At step 1 of Algorithm 3, the number of wavelet packet decomposition levels for each mode, contained in vector $\mathbf{l} = [l_1, l_2, l_3]^T$, is a required input for the computation of the wavelet packet coefficients. In [4], these values are chosen empirically. We propose an algorithm to estimate vector \mathbf{l} . To reduce the range of acceptable values, we first notice that, up to now, the interest of performing decomposition along the wavelength mode has not been emphasized when multidimensional images are processed with MWPT-MWF. That is, l_3 is always fixed to 0. Secondly, without loss of generality, there is no prior assumption concerning the structure of the images. That is, we assume there is no feature which is expressly related to rows or columns. So, the row and column dimensions are processed in the same manner, and the number of decomposition levels is equal for each dimension: $l_1 = l_2$. Thirdly, it has been shown empirically in [4] that there is no interest to perform wavelet packet decomposition with more than three levels. From the considerations above, we restrict the set of candidate instances for \mathbf{l} to $[1,1,0]^T$, $[2,2,0]^T$, and $[3,3,0]^T$.

To make our choice between these three possibilities, we propose Algorithm 4 below. In Algorithm 4, at step 1, subsampling is applied to yield a faster process. At step 2(a), we notice that, for each mode n where n is equal to 1, 2, or 3, the value of the signal subspace dimension is $K_n = \frac{l_n}{S_n \times 2^{1+l_n}}$. It is fixed to half the size of the processed wavelet packet coefficient. Notice that we do not choose optimal values of K_n in any sense at step 2(a). Indeed, we do not seek for the best denoised tensor at this step, but for the best number of decomposition levels.

Algorithm 4 Estimation of the number of decomposition levels

Input: noisy tensor R. of size $I_1 \times I_2 \times I_3$, gross estimate X 1 of R obtained by MWF with fixed rank values $K_1 = \frac{l_1}{2}, K_2 = \frac{l_2}{2}, K_3 = \frac{l_3}{2}$, set of candidate vectors \mathbf{l} .

1. Apply subsampling by factors S1, S2 and S3 along modes 1, 2 and 3, to tensor R and tensor X1,
2. For each candidate vector $\mathbf{l} = [l_1, l_2, l_3]^T$:
 - (a) apply Algorithm 3 with fixed values of the signal subspace dimensions for each decomposition level:

$$K_1 = \frac{l_1}{s_1 * 2^{1+l_1}}, K_2 = \frac{l_2}{s_2 * 2^{1+l_2}}, K_3 = \frac{l_3}{s_3 * 2^{1+l_3}}, \text{ that is, half of the size of the coefficients.}$$

(b) retrieve the error between the estimate obtained with Algorithm 3 at step 2. (a) and the subsampled version of X_1 .

3. Select the vector \mathbf{l} for which the error is minimum.

Output: optimal vector \mathbf{l} in terms of least squares error.

Still, the reference tensor which is used in Eq. (3) is obtained in a supervised way: with MWF with manually fixed ranks. In the next subsection, we propose a method to reduce the dependence of the algorithm, and to get a better reference tensor.

C. Iterative Algorithm for the Estimation of the Reference Tensor

In Eqs. (3) and (4), a reference tensor is required, which should ideally be the expected tensor itself. As we do not afford it, we propose an iterative algorithm: we first provide a gross estimate obtained with MWF, and refine in an iterative manner this reference tensor. At iteration 1 the reference tensor is the gross estimate X_1 . At iteration $r > 1$ the reference tensor is \hat{X}_{r-1} , the result of denoising obtained at the previous iteration $r - 1$. This process is repeated iteratively. The algorithm stops when the estimate at iteration r differs from the estimate at iteration $r - 1$ by a small factor δ .

In Fig. 1, we present the flowchart of the overall algorithm, with iterative estimation of the reference tensor and estimation of the ranks with a global optimization method. This overall algorithm is described in detail in Algorithm 5 below.

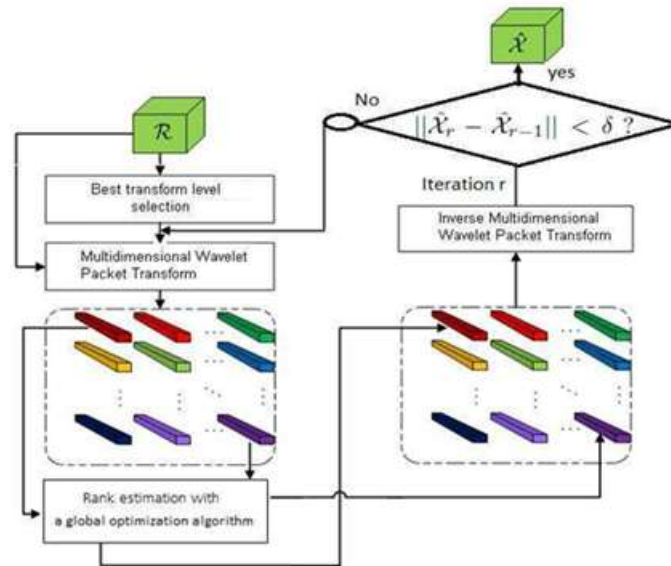


Fig. 1 Rank estimation with global optimization method in wavelet packet transform domain: iterative process

The chart in Fig. 1 corresponds to the algorithms presented above in the paper as follows:

Algorithm 5 Overall algorithm

Input: noisy tensor R , of size $I_1 \times I_2 \times I_3$

1. Obtain the gross estimate X_1 of R by MWF with fixed rank values. Apply Algorithm 4 to select the best vector \mathbf{l} with values of the number of decomposition levels for each mode
2. Set iteration $r = 1$; choose a small factor δ .
 - a) Apply Algorithm 3 (MWPT-MWF-RE) with X_1 obtained at step 1, as gross estimate. Algorithm 3 provides \hat{X}_r using the optimization method described in Algorithm 1 or Algorithm 2 for instance.
 - b) Increase the iteration index: $r = r + 1$; Set the gross estimate as \hat{X}_{r-1} , the tensor estimated at step 3. a).
 - c) Apply Algorithm 3 (MWPT-MWF-RE) with \hat{X}_{r-1} obtained at step 3. a), as gross estimate. Algorithm 3 provides \hat{X}_r .
 - d) Compute the error $e = \|\hat{X}_r - \hat{X}_{r-1}\|^2$.
 - e) If $e > \delta$, go to step 3. b). Otherwise, set $\hat{X} = \hat{X}_r$.

Output: Denoised tensor \hat{X} .

If Algorithm 5 is stopped at iteration $r = 1$, that is, at step 3. a), it is equivalent to selection of the best vector \mathbf{l} followed by Algorithm 3. In the rest of the paper, we denote Algorithm 3 as MM-GA, or MM-PSO depending on the optimization method

which is chosen. We denote Algorithm 5 as MM-Iterative. MM stands for MWPT-MWF.

V. RESULTS

A. Experimental Setup

In this section, we evaluate the performances of the proposed methods on RGB and multispectral images. We consider images where the relevant features are localized in some regions of the image. From the presence of such small local features we expect methods based on wavelet packet decomposition to provide good results, as wavelet packet decomposition permits to separate the processing of high frequency and low frequency features.

In the recent state-of-the-art about HSI denoising [13, 48], the authors emphasize the quality of the recent HSI sensors: since in most bands noise is nearly invisible for human eyes, the authors impair the images with synthetic data. In this paper, we focus on a specific application fluorescence imaging of plant leaves [49]. In the context of fluorescence imagery, the noise magnitude is inherently rather elevated, compared to the applications considered elsewhere. Indeed, the emitted fluorescence light has a much lower magnitude than the excitation light provided by a monochromator.

Consequently, we will firstly present some quantitative results obtained from artificially impaired images. These results are at first obtained from the RGB Baboon image. Secondly, they are obtained from multispectral aerial images. These multispectral images are extracted from hyperspectral images that were obtained with the ROSIS acquisition sensor over the urban area of Pavia, northern Italy. Thirdly, we present results obtained from fluorescence images which are inherently noisy. We remind that X is the noise-free tensor, R is the noised tensor and \hat{X} is the estimated tensor. The processed noised images follow the tensor model and are expressed as: $R = X + N$.

The results obtained are evaluated in terms of SNR and perceptual image quality. The perceptual image quality is measured through mean structural similarity (SSIM) [50] over all spectral bands.

We remind that $SNR = 10 \log\left(\frac{\|X\|^2}{\|X - \hat{X}\|^2}\right)$ and $SSIM(\hat{X}, X) = lu(X, \hat{X}) * co(X, \hat{X}) * st(X, \hat{X})$, where lu , co , and st are three terms aiming at image comparison: luminance, contrast, and structure.

Programmes were written in Matlab®, and executed on a PC running Windows, with a 3GHz double core and 3GB RAM.

Unless for the last considered application to fluorescence imaging, the images are artificially impaired with white, identically distributed random noise with the following input SNR (SNR_{in}) values (in dB): 5, 7.5, 10, 12.5, and 15. We present both examples for a specific value of SNR_{in} such as 5 dB or 12.5 dB, and statistical results obtained with the values of SNR_{in} cited above and 20 noise realizations.

Values of input SNR between 5 and 15 dB are realistic for the considered fluorescence images, hence our choice. The results are evaluated regarding output SNR (SNR_{out}) and output SSIM ($SSIM_{out}$). In the wavelet packet decomposition, following the recommendations in [4] we choose Coiflets as wavelet functions.

As concerns the number of decomposition levels along each mode, we set the number l_3 for the third mode through the following heuristic process: with fixed the number of decomposition levels along the first and two modes to 2, and tried several values for l_3 . We noticed that performing decomposition along the wavelength mode does improve the denoising results in terms of output SNR, at least for the considered data. Hence, we set $l_3 = 0$. We then referred to Algorithm 4 to estimate l_1 and l_2 automatically. To run Algorithm 4, we set the subsampling factor S to 4. Unless specified, its yields $l_1 = l_2 = 2$ and as optimal values. An example of the number of signal subspace dimension values to be estimated is as follows: if we obtain $\mathbf{l} = [2, 2, 0]^T$, the decomposition yields 16 wavelet packet coefficients, which are 3^{rd} -order tensors. The total number of rank values to be estimated is $3 \prod_{n=1}^3 2^{l_n} = 3 * (2^2 * 2^2 * 2^0) = 48$.

We test the proposed methods (Algorithms 3) with either genetic algorithm, particle swarm, or Nelder-Mead optimization methods. We also test the iterative method presented in section IV (Algorithms 5). These methods are denoted as MM-GA, MM-PSO, or MM-Nelder, for Algorithm 3, and MM-Iterative for Algorithm 5. We selected state-of-the-art comparative methods, based on data projection on a signal subspace: AIC estimating the subspace ranks in the wavelet framework (proposed in [4]) denoted by MM-AIC, truncation of higher order decomposition denoted by THOSVD [51] and multiway Wiener filtering denoted by MWF [8]. We omitted some other methods, based on band-by-band wavelet processing such as FoRWaRD [52], or classical linear smoothing filters because they provide either poor SSIM or output SNR values [12]. We propose Perona-Malik [18] as a comparative method. When Perona Malik is applied, the image is smoothed through a diffusion process, in such a way as to encourage intraregion smoothing rather than interregion smoothing. The Perona-Malik method is denoted as 'Per-Mal' in the tables presented further in the paper. Following the recommendations in [18], the parameters are as follows: the number of iterations is 15, the constant integration δt is fixed to its maximum value, $1/7$; the gradient modulus threshold that controls the conduction is $\kappa = 30$; and the conduction coefficient function privileges high contrast edges. The tested algorithms require a few parameters which are set once for all processed images: following the recommendations in [16] PSO is run with the swarm size is $Q = 10$ (about three times the number of unknowns which is three)

and $\epsilon = 10^{-6}$. This yields $\max_{it} = 100$ iterations. The acceleration constants γ_{1i} and γ_{2i} are set to 2 and 3 respectively. We run a version of GA using a Lagrangian algorithm [14, 15], with $Q = 300$ individuals in the initial population. We use the fitness function to provide a measure of how individuals have performed in the problem domain. When THOSVD or MWF are run, we set the values of signal subspace dimension equal to 60% of the data size along each mode. Unless specified, to create the reference tensor we set the values of signal subspace dimension equal to 60% of the data size along each mode. When the iterative process described in subsection C. is used, 3 iterations are used to refine the gross tensor estimate. Unless specified, the subsampling factors used to obtain a smaller version of the wavelet packet coefficient tensor are set to $S_1 = S_2 = 4$, and $S_3 = 2$ in algorithms 3 and 4. For the RGB display of the multispectral images throughout the section, we select 3 representative bands. Unless specified, they are in the red (690 nm), green (550 nm), and blue (450 nm) wavelength domains respectively.

B. Baboon RGB Image $256 \times 256 \times 3$

To process this RGB image, the values of signal subspace dimension in the reference tensor are set to 80% along the spatial modes, and 100% along the wavelength mode, that is, 3 because only three bands compose this image. The subsampling factors are set to $S_1=S_2=4$, and $S_3=1$. This image is processed with 2 wavelet packet decomposition levels along the spatial modes, leading to wavelet packet coefficients of size $64 \times 64 \times 3$.

1) Exemplification with $SNR_{in} = 5dB$

Table 1 provides the estimated values for the first computed detail coefficient, and for the approximation coefficient. Notice that the spatial ranks are multiples of 4, because $S_1=S_2=4$. MM-GA and MM-PSO yield similar spatial rank values, whereas AIC tends to underestimate the value of the rank along the first mode (rows), for both approximation and detail coefficients. Otherwise, considering the detail coefficient, the maximum value is obtained for rows and columns. This can be explained as follows: In the case where salient structures are present in an image, such as horizontal lines or columns, there exists a set of columns or rows which are mutually proportional, and the rank of such images is rather low. When we consider the estimated rank values for detail coefficient (see Table 1), we notice that the rank values are elevated (64 for all but one value). This is because there is no salient structure in the detail coefficients in the case of this image: it rather contains noise.

TABLE 1 BABOON RGB IMAGE $256 \times 256 \times 3$: ESTIMATED RANKS VALUES FOR THE FIRST DETAIL COEFFICIENT AND THE APPROXIMATION COEFFICIENT

Coefficient	Method		
	MM-GA	MM-PSO	MM-AIC
Approximation	(54, 64, 2)	(52, 56, 3)	(12, 64, 3)
Detail	(64, 64, 2)	(64, 64, 2)	(16, 64, 3)

Fig. 2 presents the results obtained with the proposed and comparative methods on the Baboon image. It can be noticed that Perona-Malik yields a blurred image, that the image provided by MM-AIC is rather blurred along rows. When MM-Iterative is applied, the result image is slightly more noisy than the result provided by MM-PSO, but the contrast and the contours are better preserved. This is done to the cost of a higher computational load (see Table 2).

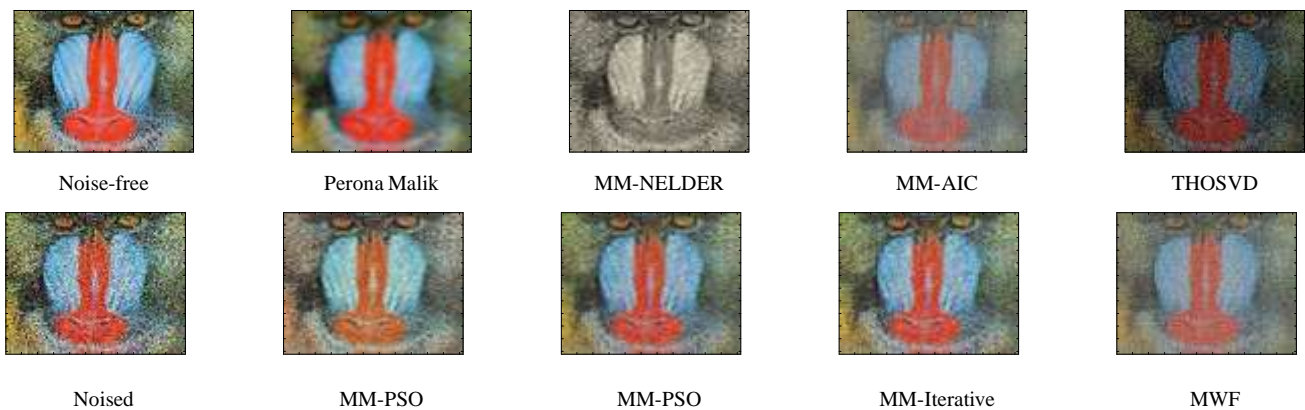


Fig. 2 Baboon RGB image $256 \times 256 \times 3$: Noise-free, noised 5 dB, denoising results

TABLE 2 BABOON RGB IMAGE $256 \times 256 \times 3$: COMPUTATIONAL LOAD ALL METHODS (IN SEC.)

Per-Mal	MM-GA	MM-Nelder	MM-PSO	MM-AIC	MM-Iterative	THOSVD	MWF
0.931	49.03	11.59	8.931	2.398	24.59	0.143	0.369

2) Statistical Results

The statistical results presented in Table 3, and Table 4 show that, for all input SNR values but 5 dB, the proposed methods MM-GA, MM-PSO, and MM-Iterative perform better than Perona-Malik, MM-AIC, MWF or truncation of HOSVD, regarding output SNR and SSIM.

Firstly, in the case where $SNR_{in} = 5dB$, we notice that the output SNR values provided by MM-AIC are significantly lower than those obtained with MM-GA or MM-PSO due to the blurred aspect of the result obtained by AIC in Fig. 2, and to the underestimation of the rank along rows. For such an image with 3 bands, the comparative Perona-Malik method is valuable for the low SNR value of 5dB but the output SNR is almost the same whatever the input SNR. Secondly, if we consider the proposed methods, and for any value of input SNR, when the iterative method is used, the output SNR and SSIM are slightly better than when PSO is used, at the expense of a larger computational load: 8.931 sec. for MM-PSO, 24.59 sec. for MM-Iterative, and 2.398 sec. for MM-AIC. When this image is considered, MM-GA is significantly slower than MM-PSO. That is why PSO is used in the iterative method, and not GA. We wish to confirm this with multispectral images with more bands.

TABLE 3 BABOON RGB IMAGE $256 \times 256 \times 3$: SNR_{out} ALL METHODS

SNR_{in}	5 dB	7,5 dB	10 dB	12,5 dB	15 dB
SNR_{out} Per-Ma	13.24	13.36	13,40	13,45	13,46
SNR_{out} MM-GA	12.57	13.87	15,05	16,44	17,64
SNR_{out} MM-Nealder	9,57	9,92	10,25	10,55	10,78
SNR_{out} MM-PSO	12.84	14,065	16,43	17,92	19,52
SNR_{out} MM-AIC	8.81	10,29	12,01	13,96	15,50
SNR_{out} MM-Iterative	13.30	14,52	16,48	17,98	19,61
SNR_{out} THOSVD	6.48	8,67	10,96	13,33	15,78
SNR_{out} MWF	10	12,07	14,14	16,04	17,93

TABLE 4 BABOON RGB IMAGE $256 \times 256 \times 3$: $SSIM_{out}$ ALL METHODS

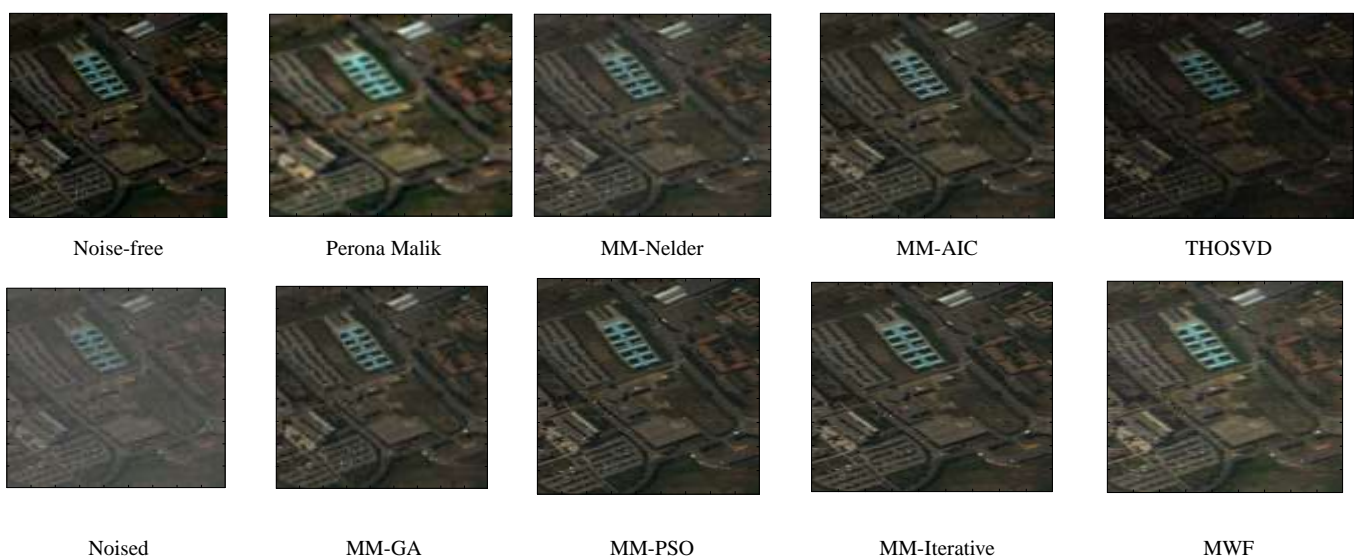
$SSIM_{in}$	0,2530	0,344	0,451	0,561	0,665
$SSIM_{out}$ Per-Ma	0,2500	0,3000	0,3000	0,3100	0,3100
$SSIM_{out}$ MM-GA	0,4671	0,5588	0,6433	0,7313	0,8038
$SSIM_{out}$ MM-Nealder	0,4359	0,5342	0,6200	0,6926	0,7507
$SSIM_{out}$ MM-PSO	0,4161	0,5357	0,6523	0,7330	0,7380
$SSIM_{out}$ MM-AIC	0,2642	0,3544	0,4516	0,5416	0,6082
$SSIM_{out}$ MM-Iterative	0,4302	0,5394	0,6575	0,7332	0,7900
$SSIM_{out}$ THOSVD	0,2946	0,3926	0,4996	0,6058	0,7039
$SSIM_{out}$ MWF	0,3488	0,4469	0,5435	0,6294	0,7116

C. Multispectral Image PAVIAU Image $512 \times 512 \times 16$

In this subsection we consider a multispectral image, of size $512 \times 512 \times 16$. This image is processed with 2 wavelet packet decomposition levels along the spatial modes, leading to wavelet packet coefficients of size $128 \times 128 \times 16$. Firstly, we exemplify the proposed and comparative methods with the case where the input $SNR_{in} = 5$ dB. Secondly, we propose an exemplification with an input SNR 12.5 dB. Thirdly, we provide statistical results obtained with various values of input SNR.

1) Exemplification with $SNR_{in} = 5dB$

Some examples of result images obtained when processing the multispectral image PaviaU of size $512 \times 512 \times 16$ are presented in Figs. 3 and 4 for the case where the input SNR is 5 dB. Fig. 4 is a zoom on the upper right region of the image.

Fig. 3 PaviaU image $512 \times 512 \times 16$: Noise-free, noised 5 dB, denoising results

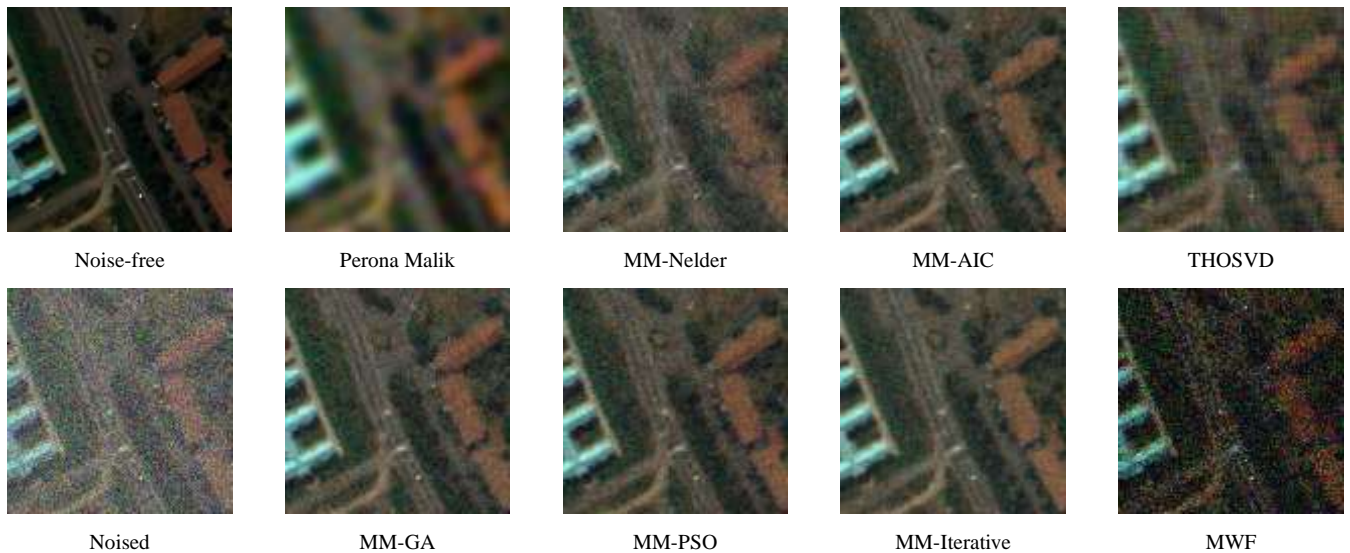


Fig. 4 PaviaU image $512 \times 512 \times 16$, zoom: Noise-free, noised 5 dB, denoising results

Table 5, which provides the processing time with an image containing 16 bands, shows that the proposed methods MM-GA, MM-PSO, and MM-Iterative are slower than MM-AIC.

TABLE 5 PAVIAU IMAGE $512 \times 512 \times 16$: COMPUTATIONAL TIME FOR EACH METHOD

Per-Mal	MM-GA	MM-Nelder	MM-PSO	MM-AIC	MM-Iterative	THOSVD	MWF
17.83	196.5	52.03	48.12	16.22	138.5	2.66	7.16

Table 6 presents results obtained regarding output SNR from the zoom image in Fig. 4, when a subset of bands is selected out of the 16 bands which compose the image. The number of selected bands varies from 12 to 3. The image is still impaired with an input SNR of 5 dB.

TABLE 6 PAVIAU IMAGE, ZOOM, SNRIN 5 DB: SNROUT FOR VARIOUS NUMBER OF BANDS

Nb Bands	SNR _{out} MM-PSO	SNR _{out} MM-AIC	SNR _{out} MM-Iterative
12	14.42	13.21	14.22
10	14.09	11.99	14.61
8	13.40	12.03	13.73
6	12.84	9.95	12.95
4	12.66	7.85	13.16
3	13.46	7.56	13.47

The numerical results in Table 6 show that the performances of **MM-PSO** and **MM-Iterative** hardly change when the number of bands is reduced. Conversely, the output SNR provided by **MM-AIC** sinks, when the number of bands is reduced from 12 to 3.

2) Exemplification with $SNR_{in} = 12,5dB$

The example in Fig. 5 concerns a case where the input SNR is 12.5 dB.



Fig. 5 PaviaU image $512 \times 512 \times 16$: Noise-free, noised 12.5 dB, denoising results

3) Statistical Results

These result images presented in this subsection yield the following comments: we notice that the result image provided by the Perona-Malik method is more blurred than all other result images. As for the methods which are not based on wavelet packet decomposition, the truncation of the HOSVD yields an image which is still rather noisy, and the MWF yields a column and row artifact and a poor detail preservation. As for the methods which include wavelet packet decomposition, the details are slightly better preserved when MM-GA, MM-PSO, and MM-Iterative are used than in the case where MM-AIC is used.

It is worth noticing that the performance of MM-GA, and MM-PSO are barely the same when the number of bands is reduced from 16 to 3. On the contrary, the performance of MM-AIC sinks when the number of bands decreases. This is coherent with the result presented in Table 3, for an input SNR of 5 dB, with the Baboon image which is composed of only 3 bands: MM-GA, and MM-PSO outperform MM-AIC in this case as well.

The numerical results in Tables 7 and 8 are by the visual aspect of the images obtained with input SNR values 5 and 12.5 dB: the output SNR and SSIM values are higher when the MWPT-MWF-RE algorithm is used, at the expense of a higher computational load.

TABLE 7 PAVIAU $512 \times 512 \times 16$: SNR_{out} ALL METHODS

SNR_{in}	5 dB	7,5 dB	10 dB	12,5 dB	15 dB
SNR_{out} Per-Ma	11,97	12,05	12,10	12,13	12,14
SNR_{out} MM-GA	15,38	17,41	18,91	20,54	21,82
SNR_{out} MM-Nealder	13,89	14,77	15,39	15,84	16,17
SNR_{out} MM-PSO	16,48	17,86	19,10	20,58	22,24
SNR_{out} MM-AIC	16,23	17,26	18,92	20,52	22,15
SNR_{out} MM-Iterative	16,40	18,00	19,24	20,79	22,29
SNR_{out} THOSVD	7,17	9,42	11,73	14,09	16,49
SNR_{out} MWF	13,47	15,00	16,57	18,18	19,84

TABLE 8 PAVIAU $512 \times 512 \times 16$: $SSIM_{out}$ ALL METHODS

$SSIM_{in}$	0,48	0,55	0,65	0,74	0,80
$SSIM_{out}$ Per-Ma	0,6539	0,6713	0,6918	0,7040	0,7076
$SSIM_{out}$ MM-GA	0,7772	0,8219	0,8823	0,9253	0,9424
$SSIM_{out}$ MM-Nealder	0,7149	0,7449	0,7554	0,7664	0,7820
$SSIM_{out}$ MM-PSO	0,8149	0,8602	0,9047	0,9264	0,9499
$SSIM_{out}$ MM-AIC	0,8021	0,8687	0,8970	0,9396	0,9530
$SSIM_{out}$ MM-Iterative	0,8280	0,8398	0,9073	0,9360	0,9439
$SSIM_{out}$ THOSVD	0,6477	0,7406	0,8179	0,8774	0,9215
$SSIM_{out}$ MWF	0,6642	0,7397	0,7915	0,8637	0,8988

From these remarks, we infer that the proposed methods MM-PSO or MM-Iterative are preferable in applications where the processed multispectral image is composed of a low number of bands and the input SNR are rather low.

D. Multispectral Image PAVIAU Image $128 \times 128 \times 16$

In this subsection, we consider a multispectral image, of size $128 \times 128 \times 16$. This image is processed with 2 wavelet packet decomposition levels along the spatial modes, leading to wavelet packet coefficients of size $32 \times 32 \times 16$. This image is processed with subsampling parameters $S1 = S2 = S3 = 1$.

1) Exemplification with $SNR_{in} = 12,5dB$

The results obtained are exemplified in Fig. 6, where the image is impaired with $SNR_{in} = 12,5dB$.

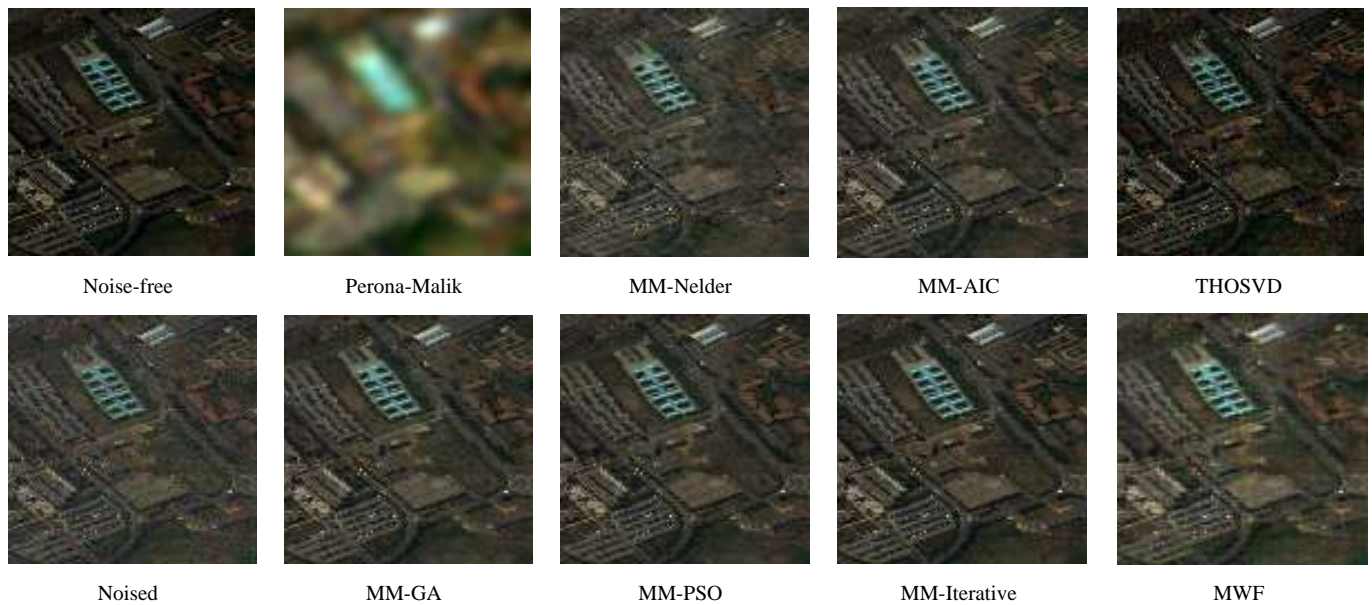


Fig. 6 PaviaU image $128 \times 128 \times 16$: Noise-free, noised 12.5 dB, denoising result

2) Statistical Results

As shown in Tables 9 and 10, for all input SNR values, the iterative method provides the best values of output SNR and output SSIM. As shown in Table 11, its computational load is about 20 sec., higher than methods which are not based on multiway Wiener filtering of wavelet packet coefficients, but still 2 times lower than in the case where MM-GA is used. The interest of using PSO in the iterative method is to reach a good compromise between the improvement of the image quality, and the computational load. The MM-Nelder method yields output SNR and SSIM values which are lower than MM-GA or MM-PSO. This can be due to the fact that, as three unknowns are expected, Nelder optimization method converges towards non-critical points. In the following, we will focus on MM-GA, MM-PSO, MM-AIC, and MM-Iterative methods.

TABLE 9 PAVIAU IMAGE $128 \times 128 \times 16$: SNR_{out} ALL METHODS

SNR_{in}	5 dB	7,5 dB	10 dB	12,5 dB	15 dB
SNR_{out} Per-Ma	8,66	8,69	8,72	8,73	8,74
SNR_{out} MM-GA	12,24	14,29	15,51	16,71	18,97
SNR_{out} MM-Nealder	10,60	11,20	11,75	12,03	12,24
SNR_{out} MM-PSO	12,65	14,36	15,40	17,44	18,95
SNR_{out} MM-AIC	13,40	15,13	17,01	18,92	20,91
SNR_{out} MM-Iterative	13,77	15,45	17,33	18,99	20,94
SNR_{out} THOSVD	6,10	8,32	10,61	12,93	15,31
SNR_{out} MWF	5,10	7,59	10,10	12,59	15,10

TABLE 10 PAVIAU IMAGE $128 \times 128 \times 16$: $SSIM_{out}$ ALL METHODS

$SSIM_{in}$	0,35	0,47	0,58	0,66	0,76
$SSIM_{out}$ Per-Ma	0,3015	0,3017	0,3171	0,3223	0,3237
$SSIM_{out}$ MM-GA	0,5959	0,7171	0,7741	0,8321	0,8891
$SSIM_{out}$ MM-Nealder	0,5179	0,5770	0,5988	0,6108	0,6351
$SSIM_{out}$ MM-PSO	0,6201	0,7233	0,7835	0,8391	0,8803
$SSIM_{out}$ MM-AIC	0,5773	0,7206	0,7612	0,8578	0,9077
$SSIM_{out}$ MM-Iterative	0,6603	0,7524	0,7978	0,8591	0,9150
$SSIM_{out}$ THOSVD	0,4508	0,5702	0,6697	0,7656	0,8389
$SSIM_{out}$ MWF	0,3554	0,4707	0,5822	0,6660	0,7621

TABLE 11 PAVIAU IMAGE $128 \times 128 \times 16$: COMPUTATIONAL LOAD ALL METHODS (IN SEC.)

Per-Mal	MM-GA	MM-Nelder	MM-PSO	MM-AIC	MM-Iterative	THOSVD	MWF
1.446	39.94	10.11	6.880	1.530	20.54	$8.99 \cdot 10^{-2}$	$1.648 \cdot 10^{-1}$

E. Subsampling and Rank Estimation

The purpose of this subsection is to evaluate the effect of subsampling on the estimation of the rank values. The multispectral image in Fig. 6 is a smaller version of the image in Fig. 5, with the same SNR of 12.5 dB. It has been obtained by subsampling along spatial modes by a factor 4, and keeping the same number of bands. We expect to obtain subspace rank

values which are four times smaller than for the large image. Table 12 presents the values of signal subspace dimensions obtained from the image in Fig. 5, and Table 13 presents the values of signal subspace dimensions obtained from the image in Fig. 6.

TABLE 12 PAVIAU $512 \times 512 \times 16$: ESTIMATED RANKS VALUES FOR THE FIRST DETAIL COEFFICIENT AND THE APPROXIMATION COEFFICIENT

Coefficient	Method		
	MM-GA	MM-PSO	MM-AIC
Approximation	(116, 116, 14)	(113, 114, 14)	(43, 43, 6)
Detail	(78, 68, 12)	(73, 72, 12)	(1, 128, 16)

TABLE 13 PAVIAU IMAGE $128 \times 128 \times 16$: ESTIMATED RANKS VALUES FOR THE FIRST DETAIL COEFFICIENT AND THE APPROXIMATION COEFFICIENT

Coefficient	Method		
	MM-GA	MM-PSO	MM-AIC
Approximation	(29, 29, 15)	(28, 30, 14)	(32, 32, 13)
Detail	(15, 20, 9)	(20, 20, 2)	(32, 32, 9)

Firstly, here is a comparative study of the rank values obtained with the large and with the smaller image: When GA or PSO are used, comparing the values in Tables 12 and 13 yields the following comments: when considering the approximation coefficients, the values along rows and columns of signal subspace dimension are 3.8 to 4.1 times higher. For the detail coefficients, the ratio between dimension values is slightly farther from 4. This is coherent with the subsampling factor between the large and the small image, equal to 4. The slight bias with respect to the subsampling factor may be due to the fact that a different noise realization impairs the image. Additionally, we performed similar tests with the iterative method. When it is used, the estimated rank values increase throughout the iterations and reach almost always the size of the wavelet packet coefficient along each mode at the last iteration. Indeed, the noise magnitude of the processed image shrinks throughout the iterations, hence the increase in the values of signal subspace dimension. These results assess, at least for the considered images and subsampling factors, the relevance of step 3a in Algorithm 3.

Secondly, here are comments which aim at comparing the rank values in either approximation or detail coefficients: When referring to the results presented in Table 12 and 13, we notice that AIC always provides the maximum possible value, except once: the least possible value, that is, 1, is provided for a detail coefficient of the image in Fig. 4. The interest of this method is therefore limited in this case. When GA or PSO are used, the rank values for the approximation coefficient is more elevated than the rank values for the detail coefficient. This is due to the ability of wavelet packet decomposition to concentrate the high frequency components, that is, the noise, in the detail coefficients. The noise magnitude is less elevated in the approximation coefficient, and a larger proportion of singular vectors are considered as forming part of the signal subspace. So the value of the signal subspace dimension is more elevated for the approximation coefficient than for the detail coefficient.

F. Multispectral Image PAVIAU Image $128 \times 128 \times 32$

The results presented above have shown the interest of using MM-PSO or MM-Iterative to reach elevated output SNR values when the number of bands in the processed multispectral image is rather low. As their computational load is more elevated than MM-AIC, we wish to provide an alternative for the case where the multispectral image is composed of a high number of bands. We aim at demonstrating the interest of fixed point algorithm when the processed image is large, and impaired with a low input SNR of 5 dB. The visual results obtained are presented in Fig. 7: we display the noise-free image, the noised image impaired with a 5 dB white Gaussian noise, and the denoising results obtained with MM-PSO and MWF using either singular value decomposition (SVD) or fixed point (FP) algorithm.

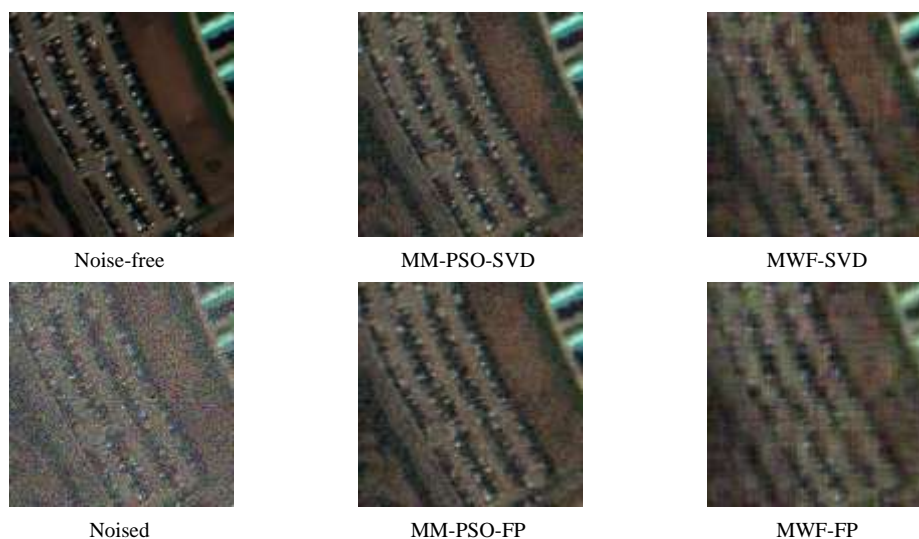
Fig. 7 PaviaU image $128 \times 128 \times 32$: Noise-free, noised 5 dB, results with PSO and MWF with either SVD or Fixed Point

Table 14 provides the rank values estimated by GA, PSO and AIC. It shows that, compared to the case when the input SNR is 12.5 dB (see Table 13), the estimated rank values are smaller, even much smaller for the detail coefficients. Table 15 presents the computational loads obtained with GA, PSO, the iterative method, and MWF when either SVD, or FP, are used. The computational time required by AIC in the same conditions is 2.77 sec.

TABLE 14 PAVIAU IMAGE $128 \times 128 \times 32$: ESTIMATED RANKS VALUES FOR THE FIRST DETAIL COEFFICIENT AND THE APPROXIMATION COEFFICIENT

Coefficient	Method		
	MM-GA	MM-PSO	MM-AIC
Approximation	(26, 26, 18)	(16, 17, 11)	(24, 32, 15)
Detail	(4, 4, 4)	(13, 8, 5)	(1, 32, 12)

TABLE 15 PAVIAU MULTISPECTRAL IMAGE $128 \times 128 \times 32$: COMPUTATIONAL TIME FOR EACH METHOD WITH EITHER SVD OR FIXED POINT

	MM-GA	MM-PSO	MM-Iterative	MWF
SVD	262.16	54.47	259.48	0.29
FP	258.92	29.55	129.76	0.23

Table 16 presents output SNR and SSIM values obtained with MM-PSO when either SVD, or FP, are used, and with MM-AIC.

TABLE 16 PAVIAU IMAGE $128 \times 128 \times 16$, NOISED 5 DB: SNR_{out} AND $SSIM_{out}$ OBTAINED BY MM-PSO-SVD, MM-PSO-FP, MM-AIC

	MM-PSO-SVD	MM-PSO-FP	MM-AIC
SNR_{out}	13.14	12.58	12.17
$SSIM_{out}$	0.62	0.56	0.58

In the conditions presented above, we notice that for this image fixed point algorithm divides the computational load by almost 2 when MM-PSO is used (see Table 15), at the expense of slightly smaller output SNR and SSIM values (see Table 16). From Fig. 7, we infer that using FP instead of SVD degrades the details. The output SNR value is still higher than when MM-AIC is used.

G. Performance and Number of Bands

In this section, we study the performance of MM-AIC and MM-PSO for multispectral images with various values for the number of bands. For this, we have selected CAVE database of multispectral images [53]. This database is composed of images of size $512 \times 512 \times 31$. That is, they contain 31 bands, regularly obtained from the wavelength range 400 to 700 nm, with a 10 nm step between each band.

We considered the image called “jelly beans” because it contains lots of small features and details to be preserved. We exemplify the MM-AIC and MM-PSO with two images impaired with random noise in such a way that $SNR_{in} = 2$ dB. The first (resp. second) image contains 5 (resp. 31) bands regularly spaced between 400 and 700 nm.

1) Exemplification with $SNR_{in} = 2$ dB and 5 Bands

The image presented in Fig. 8 contains 5 bands. The MM-AIC method provided $SNR_{out} = 5,0$ dB, and the MM-PSO method provided $SNR_{out} = 8,8$ dB.



Noise-free



MM-AIC

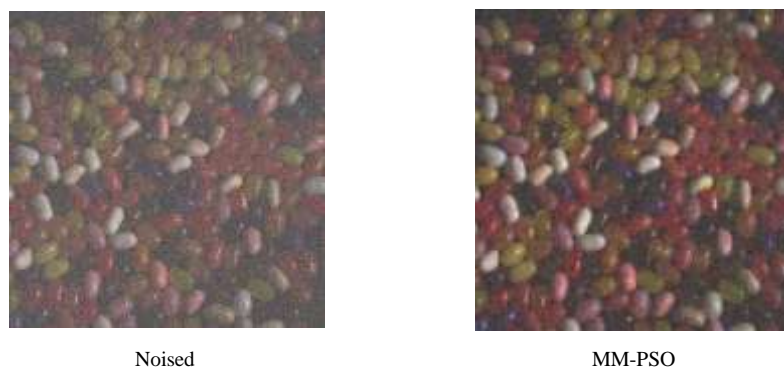


Fig. 8 “Jelly bean” multispectral image, 5 bands: noise-free and noised images; result obtained by MM-AIC and MM-PSO

2) Exemplification with $SNR_{in} = 2$ dB and 31 Bands

The image presented in Fig. 9 contains 5 bands. The MM-AIC method provided $SNR_{out} = 21,7$ dB, and the MM-PSO method provided $SNR_{out} = 24,5$ dB.

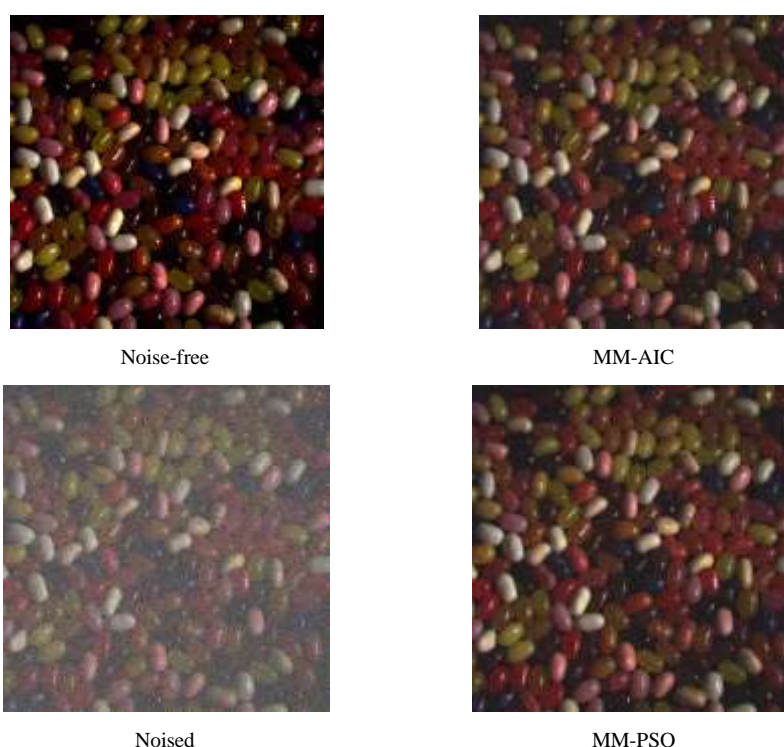


Fig. 9 “Jelly bean” multispectral image, 31 bands: noise-free and noised images; result obtained by MM-AIC and MM-PSO

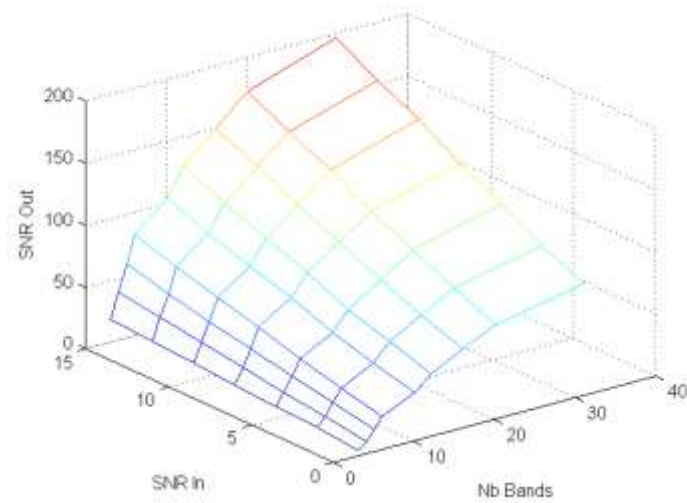
As comments to Figs. 8 and 9, we first can notice that the higher the number of bands, the better the aspect of the denoised image, with either MM-AIC and MM-PSO. Secondly, we notice that the result obtained by MM-PSO in terms of output SNR is better than the result obtained with MM-AIC. The difference is slightly higher (3,8 dB) when a low number of bands (5) is processed. When the whole “jelly beans” image is processed, with all 31 bands, the difference is 2,8 dB. As stated in section C for another multispectral image, the proposed MM-PSO method behaves better than MM-AIC for multispectral images with a low number of bands. Indeed, AIC is a statistical criterion whose performances are undoubtedly improved when a large amount of data are available.

3) Results for Various Values of Input SNR and Number of Bands

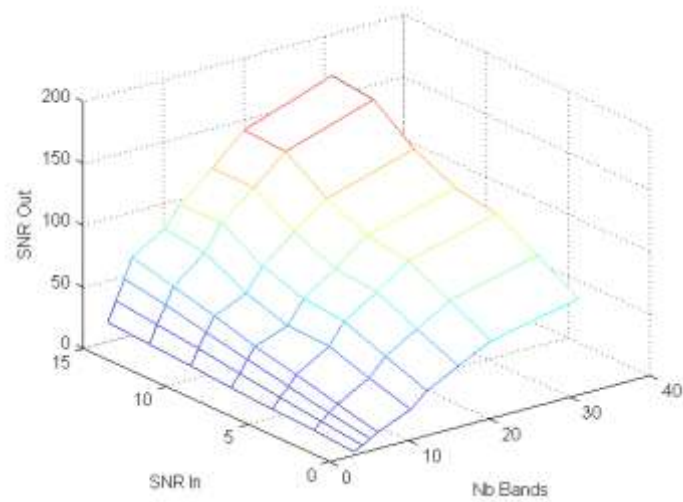
We now present results obtained with the following values of input SNR: 0, 2.5, 5, 7.5, 10, 12.5, and 15 dB, and the following values for the number of bands: 3, 4, 5, 6, 10, 12, 16, 20, and 31. These bands are selected out of the 31 bands of the “Jelly bean” multispectral image [53], and regularly spaced between 400 and 700 nm. For instance, when 3 bands are selected, they correspond to 400 nm, 550 nm, and 700 nm. When 31 bands are selected, all bands between 400 and 700 nm with a 10 nm spacing are included in the processed image.

Fig. 10 presents the results obtained in terms of output SNR for all couples of input SNR and number of bands: Fig. 10 a)

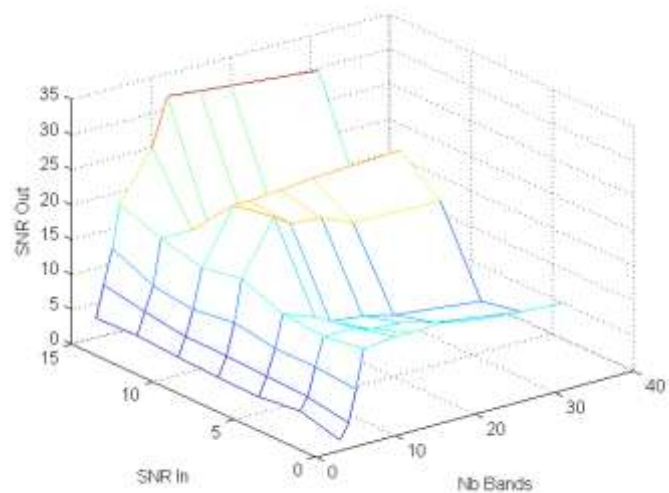
shows the results obtained with MM-PSO, Fig. 10 b) the results obtained with MM-AIC, and Fig. 10 c) shows the difference between the values obtained with MM-PSO and MM-AIC. For both methods, the larger the number of bands, the better the result obtained.



a)



b)



c)

Fig. 10 Output SNR for various values of input SNR and number of bands: a) MM-PSO, b)MM-AIC; c) Difference MM-PSO - MM-AIC

We notice that the difference is always positive, which means that the performance of MM-PSO is better in all cases. The largest difference is obtained for input SNR values between 7.5 and 15 dB, and for the number of bands between 6 and 25.

H. Plant Leaf Fluorescence Image

We now consider plant leaf images. While illuminated with a 400 nm light source, a plant leaf emits fluorescence light of little energy, in a wavelength range between 600 and 800 nm. Because of the low brightness of the light emitted through the fluorescence phenomenon, the gain of the imaging sensors are pushed to a high value, and the exposure time must be elevated to capture this phenomenon. The electronic noise is then of large magnitude, and the acquired images are noisy.

The image acquisition setup that was used is as follows: a hyperspectral camera HySpex VNIR-1600 (Norsk Elektro Optikk, Norway) acquired successive lines of 1600 pixels and 160 spectral bands ranging from 415 to 994 nm with a 3.7 nm spectral sampling interval. The lighting was provided by halogen source positioned close to the cameras. The illumination zenith angles were set to $\theta = 20^\circ$. A monochromator is placed in the incoming halogen irradiance between the leaf and the Halogen. This monochromator permits to select the wavelength which induces the fluorescence phenomenon. Fig. 11 presents, for a relevant pixel of the acquired scene, the spectrum acquired by the camera. It focuses on the relevant part of the spectrum, corresponding to the fluorescence light, located between 600 and 800 nm.

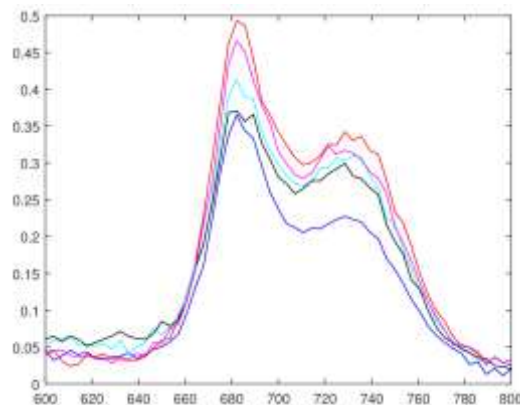


Fig. 11 Leaf fluorescence, wavelength range of interest

If the time allowed for the acquisition of each frame is lowered, the electronic noise increases with respect to the relevant part of the data. The signal to noise ratio sinks, hence the interest of denoising.

Fig. 12 shows the image obtained in the best conditions, with an elevated power for the incident light, and 50 ms of exposure time. We display a subimage of size 256×256 in Fig. 12 a), and two other subimages of size 128×128 and 64×64 in Figs. 12 b) and 12 c).

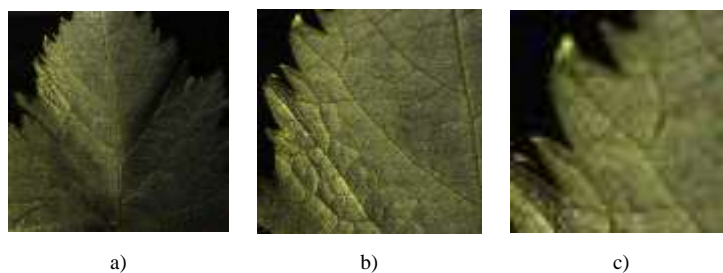


Fig. 12 Leaf fluorescence image: a) large image; b) zoom 1; c) zoom 2

Figs. 13, 14, 15, and 16 correspond to different values of exposure time between 50 ms and 250 ms, and incident power of the excitation light, for the same leaf. For a lower power of the excitation light, the setup is cheaper, but the noise level is higher.



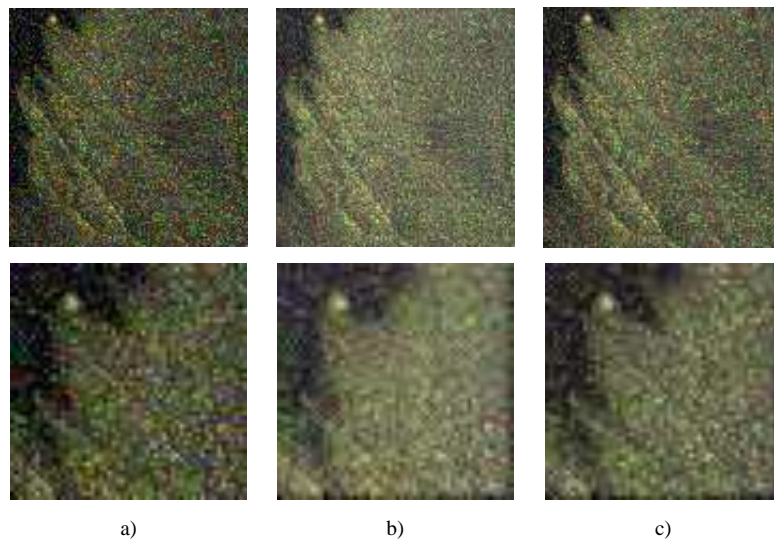


Fig. 13 Leaf fluorescence image: a) raw acquisition, SNR 0.5 dB; b) MM-PSO; c) MM-Iterative

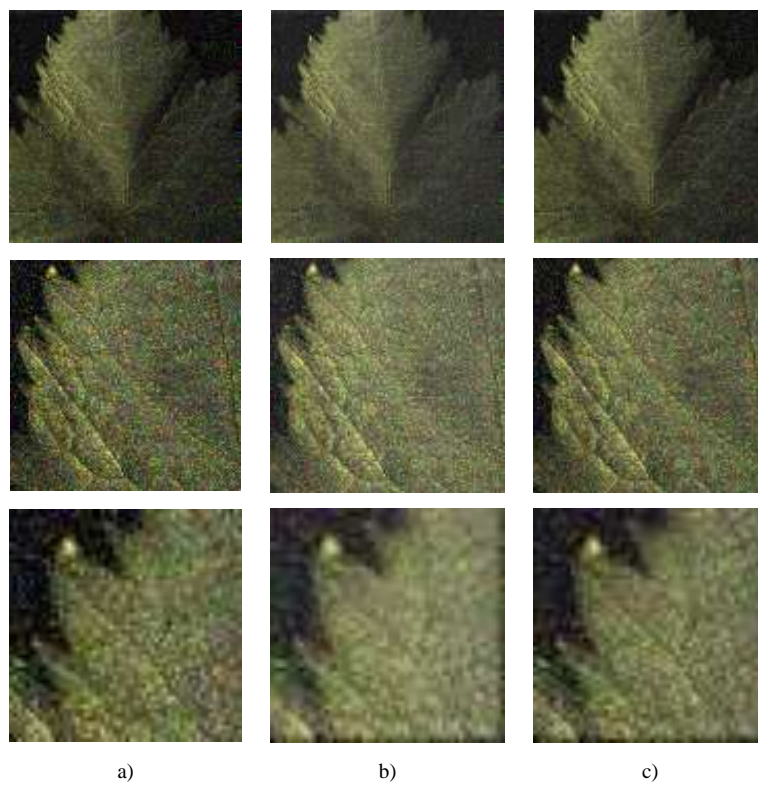


Fig. 14 Leaf fluorescence image: a) raw acquisition, SNR 5.6 dB; b) MM-PSO; c) MM-Iterative



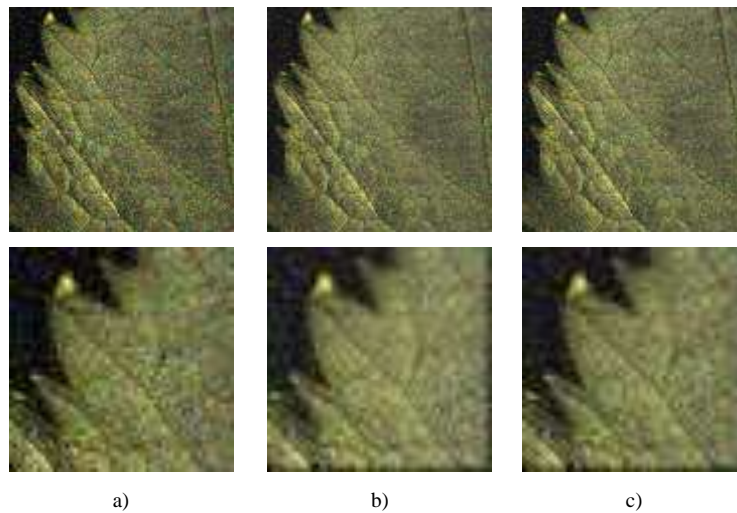


Fig. 15 Leaf fluorescence image: a) raw acquisition, SNR 10.2 dB; b) MM-PSO; c) MM-Iterative

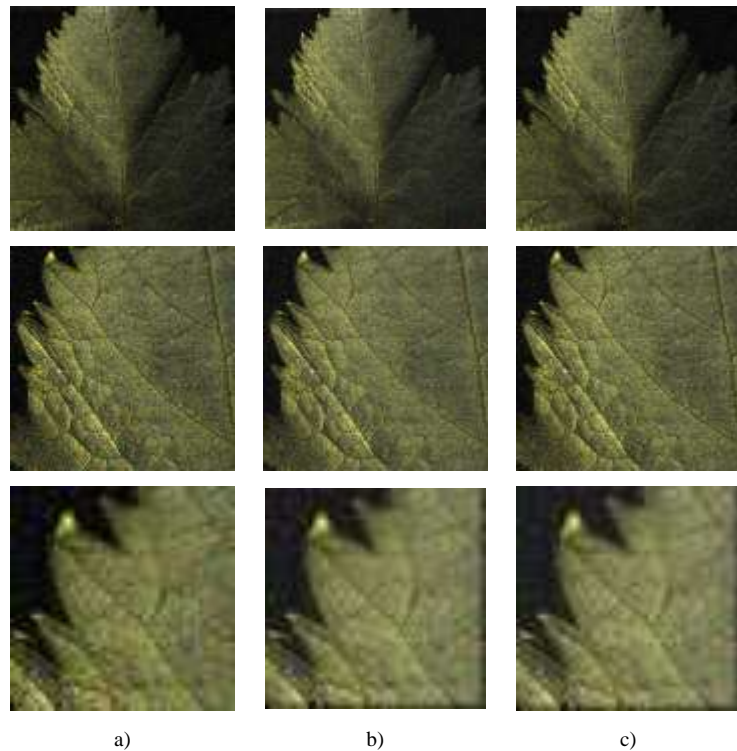


Fig. 16 Leaf fluorescence image: a) raw acquisition, SNR 15.5 dB; b) MM-PSO; c) MM-Iterative

To process these images, we select 16 wavelengths between 677 nm and 740 nm. For the representation of the acquired and processed images, we select 677, 681, and 726 nm for the R, G, and B bands.

To evaluate the denoising results quantitatively, we consider the images in Fig. 12 as noise-free. With respect to these images, the SNR of the raw images in Figs. 13, 14, 15, and 16 is 0.5, 5.6, 10.2, and 15.5 dB. The MM-PSO method yields 4.0, 8.5, 12.5, and 15.1 dB as output SNR. The MM-iterative method yields 3.6, 8.3, 12.6, and 16.3 as output SNR. The computational time for the largest image of size $256 \times 256 \times 16$ is of 13.8 sec for MM-PSO, and 33.6 sec for MM-Iterative. As for comparison, with an exposure time of 50 ms, the acquisition time required for such an image is of several minutes. We can notice that the results obtained with MM-PSO are more blurred and noisy than when MM-Iterative is applied. Indeed, MWF provides the first gross estimate. It is blurred, with low noise magnitude. When MM-PSO is run once, a denoised image is obtained where the contours are better preserved. It is used a refined reference in MM-Iterative. To get close to this new reference, the PSO algorithm converges to triplet of rank values K_1 , K_2 , and K_3 which are more elevated than when MM-PSO is used. That is why the result of MM-Iterative appears more noisy but less blurred than the result of MM-PSO.

From a qualitative point of view, our methods for denoising and detail preservation remove part of the noise while preserving the contours, that is, the frontier between ribs and limb of the leaf.

VI. DISCUSSION

While comparing the results obtained with the 'Baboon' RGB image and the multispectral images of various sizes, we notice that the denoising performance is best when the largest image is considered. When the proposed MM-Iterative method is used, on the large image of Fig. 3, the SNR improvement is as follows: an input SNR of 5 (resp. 15) dB yields an output SNR of 16 (resp. 22) dB. When the four times smaller multispectral image of Fig. 6 is processed with MM-Iterative, the ranks are estimated in proportion, four times smaller as for the larger image. However, the SNR improvement is lower: an input SNR of 5 (resp. 15) dB yields an output SNR of 14 (resp. 21) dB. We compared the results obtained with the Perona-Malik method [18], which is not based on a projection onto the signal subspace. We notice that Perona-Malik is relevant only for low input SNR values: it induces an image smoothing which degrades too much the contours.

As a partial conclusion from these results, we can infer that, when a large image is processed, it is relevant to estimate the values of signal subspace dimension on a subsampled version of the wavelet packet coefficients. This means that parameters S_1 , S_2 , S_3 in Algorithms 3 and 4 are larger than 1. In this way, the rank values are estimated on data with small size values. After the estimation on these small data, the ranks which are thereby obtained are multiplied by factors S_1 , S_2 , and S_3 , and these last rank values are used to perform denoising. This yields a smaller computational load while preserving the denoising results obtained eventually from the original (not subsampled) wavelet packet coefficients. Compared to MM-AIC, the advantage of MM-PSO and MM-Iterative is that their performances do not degrade that much when the number of bands in the processed multispectral image decreases.

To reduce the computational load, the fixed point algorithm may also be used, when the noise impairing the image is of high magnitude, and the number of bands is elevated: as shown in the experiment performed with the image in Fig. 7, a low SNR value yields small values of signal subspace dimension. This shows the interest of fixed point algorithm: it estimates only the singular vectors which belong to the signal subspace, whereas the classical singular value decomposition estimates all singular vectors exhaustively. From additional experiments we have made, we could conclude that a drawback of this method is that it should be restricted to low SNR conditions, and that the output SNR may be slightly lower than in the case where SVD is used.

As concerns the acquisition and processing of fluorescent plant leaves, the SNR is increased by about 4 dB for low input SNR values (0, 5, 10) and the method breaks down when the input SNR is 15 dB. As further work, the noise statistics of such images should be further analyzed. With the proposed application to the denoising of fluorescence images acquired from plant leaves, we make life science and tensor signal processing closer to each other. The strength of the proposed strategy is that it can be easily adapted to any other image denoising algorithm depending on a set of parameters to be tuned. In the case of the considered methods based on wavelet packet decomposition and subspace projection, and when multispectral images are processed, it is particularly interesting to work on a subsampled version of the processed image to estimate adequate parameters for denoising. Moreover, the proposed optimization methods can be adapted to any method depending on several parameters. With the proposed optimization strategy, a multi-dimensional space can be searched, yielding optimal multiplets of parameters in the sense of the criterion chosen by the user.

VII. CONCLUSION

In this paper, an algorithm named MWPT-MWF-RE has been proposed to estimate multiple values of signal subspace dimension, to apply multiway Wiener filtering in a wavelet packet framework. The final purpose of this method is to remove noise from multidimensional images.

The main advances brought by our paper are as follows: Firstly, we propose a criterion depending on signal subspace ranks, and adapt an optimization algorithm to minimize this criterion. Mainly, two bio-inspired optimization algorithms are compared: a genetic algorithm (GA) and particle swarm optimization (PSO). As PSO offers the best compromise between computational load and denoising quality, we insert it in an iterative algorithm. In the iterative algorithm, the raw estimate required in the criterion to be minimized is the estimate provided by MM-PSO obtained at the previous step.

Still, the number of decomposition levels should be estimated, and the computational load of the overall algorithm should be reduced. That is why, secondly, we propose to estimate the number of decomposition levels and the adequate values of signal subspace ranks on a subsampled version of the data: we performed tests on various types of images and concluded that, at least for the considered data, the values of the optimal signal subspace dimension in terms of denoising accuracy are divided by the subsampling factor when such a subsampling is applied to the image. Thirdly, we evaluate the performance of the proposed MWPT-MWF-RE algorithm on various multidimensional images: the RGB Baboon image, and various multispectral images extracted from the well-known Pavia University hyperspectral image. We considered an application of plant fluorescence imaging. The emitted light in this context is of very low intensity, yielding images which are impaired with rather low SNR values. A comparative study was performed involving three main types of algorithms: firstly, the diffusion-based Perona-Malik method; secondly, the truncation of HOSVD and MWF; thirdly, a method based on wavelet packet decomposition and MWF, where the dimensions of signal subspace are estimated with AIC. From the analysis and the comparative study of other similar methods, it can be concluded that an iterative algorithm with automatic rank estimation with particle swarm optimization offers a good compromise between computational load, SNR, and SSIM improvement. The

proposed method performs best, compared to AIC, when the noise magnitude is elevated and the number of bands is between 6 and 25.

As further works, we could adopt the proposed optimization strategy to any other denoising methods which require the tuning of parameters, but also to classification methods, such as for the choice of optimal kernel parameters for support vector machine.

ACKNOWLEDGMENTS

We would like to thank the Bayerische Forschungsstiftung for its financial support. We are also very grateful to Dr. Ryad Bendoula from IRSTEA Laboratory, for putting the multispectral fluorescence setup at our disposal, and for fruitful discussions.

REFERENCES

- [1] Ante Jukic, Ivica Kopriva, and Andrzej Cichocki, "Noninvasive diagnosis of melanoma with tensor decomposition-based feature extraction from clinical color image," *Biomedical Signal Processing and Control*, vol. 8(6), pp. 755-763, 2013.
- [2] Robert O. Green, Charles M. Sarture, Christopher J. Chovit, Jessica A. Faust, Pavel Hajek, and H. Lan Novak, "Aviris: A new approach to earth remote sensing," *Opt. Photon. News*, vol. 6(1), pp. 30-33, Jan. 1995.
- [3] S Holzwarth, A Muller, M Habermeyer, R Richter, A Hausold, S Thiemann, and P Strobl, "Hysens-dais 7915/rosis imaging spectrometers at dlr," In *Proceedings of the 3rd EARSel Workshop on Imaging Spectroscopy*, pp. 3-14, 2003.
- [4] T. Lin and S. Bourennane, "Hyperspectral image processing by jointly filtering wavelet component tensor," *IEEE Trans. Geosci. Remote Sens.*, vol. 51(6), pp. 3529-3541, 2013.
- [5] X. Liu, S. Bourennane, and C. Fossati, "Nonwhite noise reduction in hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 9(3), pp. 368-372, 2012.
- [6] J.P. Kerekes and J.E. Baum, "Full spectrum spectral imaging system analytical model," *IEEE Trans. on Geosc. and Remote Sensing*, vol. 43(3), pp. 571-580, Mar. 2005.
- [7] Nynke S. van den Berg, Tessa Buckle, Gijs H. KleinJan, Henk G. van der Poel, and Fijs W.B. van Leeuwen, "Multispectral fluorescence imaging during robot-assisted laparoscopic sentinel node biopsy: A first step towards a fluorescence-based anatomic roadmap," *European Urology*, 8 pages, 2016.
- [8] D. Muti and S. Bourennane, "Multidimensional filtering based on a tensor approach," *Signal Processing Journal, Elsevier*, vol. 85(12), pp. 2338-2353, Dec. 2005.
- [9] N. Renard, S. Bourennane, and J. Blanc-Talon, "Denoising and dimensionality reduction using multilinear tools for hyper-spectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 5(2), pp. 138-142, 2008.
- [10] Tao Lin, Julien Marot, and Salah Bourennane, "Advanced Concepts for Intelligent Vision Systems," *15th International Conference, ACIVS 2013*, Poznan, Poland, October 28-31, 2013. Proceedings, chapter Small Target Detection Improvement in Hyperspectral Image, pp. 460-469, Springer International Publishing, Cham, 2013.
- [11] J. Marot and S. Bourennane, "Recent advances on tensor models and their relevance for multidimensional data processing," In *2014 48th Asilomar Conference on Signals, Systems and Computers*, pp. 586-590, Nov. 2014.
- [12] Julien Marot and Salah Bourennane, "Advanced Concepts for Intelligent Vision Systems," *16th International Conference, ACIVS 2015*, Catania, Italy, October 26-29, 2015. Proceedings, chapter Improvement of a Wavelet-Tensor Denoising Algorithm by Automatic Rank Estimation, pp. 779-790, Springer International Publishing, Cham, 2015.
- [13] X. Liu, S. Bourennane, and C. Fossati, "Denoising of hyperspectral images using the parafac model and statistical performance analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50(10), pp. 3717-3724, Oct. 2012.
- [14] David E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, 1st ed., Addison-Wesley Professional, 432 pages, Jan. 1989.
- [15] Andrew R Conn, Nicholas IM Gould, and Philippe Toint, "A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds," *SIAM Journal on Numerical Analysis*, vol. 28(2), pp. 545-572, 1991.
- [16] J. Kennedy and R. Eberhart, "Particle swarm optimization," In *IEEE International Conference on Neural Networks*, pp. 1942-1948, Perth, 1995.
- [17] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E. Wright, "Convergence properties of the nelder-mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9(1), pp. 112-147, 1998.
- [18] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12(7), pp. 629-639, July 1990.
- [19] Damien Muti, Salah Bourennane, and Julien Marot, "Lower-rank tensor approximation and multiway filtering," *SIAM Journal on Matrix Analysis and Applications*, vol. 30(3), pp. 1172-1204, 2008.
- [20] L.C. Parra, C. Spence, P. Sajda, A. Ziehe, and K.R. Muller, "Unmixing hyperspectral data," *Advances in Neural Information Processing Systems*, vol. 12, pp. 942-948, 2000.
- [21] D. Letexier, S. Bourennane, and J. Blanc-Talon, "Main flattening directions and quadtree decomposition for multi-way wiener filtering," *IEEE Signal, Image and Video Processing*, vol. 1(3), pp. 253-256, 2007.
- [22] M. Wax and T. Kailath, "Detection of signals information theoretic criteria," *IEEE Trans. Acoustics Speech, and Signal Processing*, vol. 33(2), pp. 387-392, Apr. 1985.

- [23] H. Malcolm Hudson and Thomas C.M. Lee, "Maximum likelihood restoration and choice of smoothing parameter in deconvolution of image data subject to Poisson noise," *Computational Statistics and Data Analysis*, vol. 26(4), pp. 393-410, 1998.
- [24] H Lanteri, M Roche, O Cuevas, and C Aime, "A general method to devise maximum-likelihood signal restoration multiplicative algorithms with non-negativity constraints," *Signal Processing*, vol. 81(5), pp. 945-974, 2001.
- [25] Artyom M. Grigoryan, Edward R. Dougherty, and Sos S. Agaian, "Optimal Wiener and homomorphic filtration: Review," *Signal Processing*, vol. 121, pp. 111-138, Apr. 2016.
- [26] Julien Marot, Caroline Fossati, and Salah Bourennane, "About Advances in Tensor Data Denoising Methods," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, pp. 235-257, 2008.
- [27] Donald R. Jones, Cary D. Perttunen, and Bruce E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Application*, vol. 79(1), pp. 157-181, Oct. 1993.
- [28] M. Mitchell, *An introduction to genetic algorithms*, ISBN: 9780262631853, p. 221, First MIT Press, Mar. 1998.
- [29] Fuchang Gao and Lixing Han, "Implementing the Nelder-Mead simplex algorithm with adaptive parameters," *Computational Optimization and Applications*, vol. 51(1), pp. 259-277, 2010.
- [30] Russ C Eberhart, James Kennedy, et al., "A new optimizer using particle swarm theory," In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, vol. 1, pp. 39-43, New York, 1995.
- [31] William T Reeves, "Particle systems—a technique for modeling a class of fuzzy objects," *ACM Transactions on Graphics (TOG)*, vol. 2(2), pp. 91-108, 1998.
- [32] Kim-Fung Man, Kit Sang TANG, and Sam Kwong, *Genetic Algorithms: Concepts and Designs*, Springer Science & Business Media, London, 2012.
- [33] Maofu Liu, Ya Liu, Huijun Hu, and Liqiang Nie, "Genetic algorithm and mathematical morphology based binarization method for strip steel defect image with non-uniform illumination," *Journal of Visual Communication and Image Representation*, vol. 37, pp. 70-77, May 2016.
- [34] N. Mohananthini and G. Yamuna, "Comparison of multiple watermarking techniques using genetic algorithms," *Journal of Electrical Systems and Information Technology*, vol. 3, iss. 1, pp. 68-80, May 2016.
- [35] Payel Ghosh, Melanie Mitchell, James A. Tanyi, and Arthur Y. Hung, "Incorporating priors for medical image segmentation using a genetic algorithm," *Neurocomputing*, vol. 195, iss. 1, pp. 181-194, Feb. 2016.
- [36] Dominic Williams, Yalin Zheng, Pinakin Guntant Davey, Fangjun Bao, Meixiao Shen, and Ahmed Elsheikh, "Reconstruction of 3d surface maps from anterior segment optical coherence tomography images using graph theory and genetic algorithms," *Biomedical Signal Processing and Control*, vol. 25, pp. 91-98, Mar. 2016.
- [37] Chia-Feng Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34(2), pp. 997-1006, Apr. 2004.
- [38] X.H. Shi, Y.C. Liang, H.P. Lee, C. Lu, and L.M. Wang, "An improved GA and a novel PSO-GA-based hybrid algorithm," *Information Processing Letters*, vol. 93(5), pp. 255-261, Mar. 2005.
- [39] Jyoti Sharma and Ravi Shankar Singhal, "Comparative research on genetic algorithm, particle swarm optimization and hybrid GA-PSO," In *Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on*, pp. 110-114, IEEE, 2015.
- [40] Chun-Yan Liu, Cheng-Ming Zou, and Pei Wu, "A Task Scheduling Algorithm Based on Genetic Algorithm and Ant Colony Optimization in Cloud Computing," *7th International Conference on IEEE*, pp. 68-72, Nov. 2014.
- [41] Parham Moradi and Mozghan Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Applied Soft Computing*, vol. 43, pp. 117-130, June 2016.
- [42] Yong-Feng Dong, Jun-Hua Gu, Na-Na Li, Xiang-Dan Hou, and Wei-Li Yan, "Combination of genetic algorithm and ant colony algorithm for distribution network planning," In *Machine Learning and Cybernetics, 2007 International Conference on*, vol. 2, pp. 999-1002, IEEE, 2007.
- [43] Dalila Cherifi, Imane Hafnaoui, and Amine Nait Ali, "Multimodal score-level fusion using hybrid ga-pso for multibiometric system," *Informatica*, vol. 39, pp. 209-216, 2005.
- [44] H. Schwefel and G. Rudolph, *Contemporary Evolution Strategies*, Taylor and Francis Group, pp. 891-907, Berlin, 1995.
- [45] J.H. Holland, *Adaptation in Natural and Artificial Systems*, ISBN 9780262082136, 9780262581110, p. 211, The MIT Press, Apr. 1992.
- [46] Shisanu Tongchim, "Coarse-grained parallel genetic algorithm for solving the timetable problem," In *Proceedings of 3rd Annual National Symposium on Computational Science and Engineering (ANSCSE)*, Bangkok, Thailand, pp. 345-353, Jan. 1999.
- [47] Y. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1(3), pp. 211-218, 1936.
- [48] Q. Li, H. Li, Z. Lu, Q. Lu, and W. Li, "Denoising of hyperspectral images employing two-phase matrix decomposition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7(9), pp. 3742-3754, Sept. 2014.
- [49] Colm D. Everard, Moon S. Kim, and Hoyoung Lee, "A comparison of hyperspectral reflectance and fluorescence imaging techniques for detection of contaminants on spinach leaves," *Journal of Food Engineering*, vol. 143(6), pp. 139-145, 2014.
- [50] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13(4), pp. 600-612, 2004.
- [51] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(R1, R2, . . . , RN) approximation of higher-order tensor," *SIAM Journal on Matrix Analysis and Applications*, vol. 21(4), pp. 1324-1342, Apr. 2000.
- [52] R. Neelamani, H. Choi, and R. Baraniuk, "Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems," *IEEE Trans. on Signal Processing*, vol. 52(2), pp. 418-433, Feb. 2004.
- [53] CAVE multispectral image database, available at <http://www.cs.columbia.edu/CAVE/databases/multispectral/>.