4DCAF: A Temporal Approach for Denoising Hyperspectral Image Sequences

Blanca Priego[a,*], Richard J. Duro[a], Jocelyn Chanussot[b]

[a] *Integrated Group for Engineering Research, Universidade da Coruña, Spain*
[b] *GIPSA-lab, Signal & Image Dept., Grenoble Institute of Technology, Grenoble, France*

Abstract

As a consequence of the fast development of sensor technology in the last decade, it is now possible to acquire sequences of hyperspectral images at reasonable frame rates. However, these sequences may be significantly corrupted by noise, especially when the spectral coverage of the data reaches the thermal domain. While there is an abundant literature on denoising of (standard) video sequences or denoising of (still) hyperspectral images, very little has been published on denoising hyperspectral sequences. This paper presents a novel denoising method for actual hyperspectral sequences. The approach is based on spatio-spectral-temporal cellular automata-based filtering. It presents several advantages, especially the fact that the cellular automaton used is able to contemplate information concerning the type of noise present through the use of specific sequences to tune the algorithm. It also considers temporal information by means of a spatio-temporal neighborhood when processing each pixel of the sequence. The proposed method outperforms several state-of-the-art algorithms on both simulated and real sequences.

Keywords: Hyperspectral, temporal denoising, cellular automata, 4DCAF

*Corresponding author
Email addresses: blanca.priego@udc.es (Blanca Priego), richard@udc.es (Richard J. Duro), jocelyn.chanussot@gipsa-lab.grenoble-inp.fr (Jocelyn Chanussot)

1. Introduction

This work deals with the preprocessing of hyperspectral image sequences as a previous step to other applications such as segmentation or classification tasks. In particular, the method proposed here is aimed at noise filtering or denoising of sequences that are significantly corrupted by noise. This problem is very common in some applications, especially when the spectral coverage of the data reaches the thermal domain.

Denoising or restoration of images is one of the most researched topics within the image processing field in the last two decades. Most of the algorithms that have been developed are focused on processing single images (2-D approaches) (Rudin et al., 1992; Portilla et al., 2003; Buades et al., 2005; Aharon et al., 2006; Luisier and Blu, 2008; Dabov et al., 2006), on denoising video sequences (Dabov et al., 2007; Priego et al., 2013; Maggioni et al., 2013) or still hyperspectral images (Peng et al., 2014; Renard et al., 2008; Liu et al., 2012; Lam et al., 2012; Liao et al., 2015; Salmon et al., 2014; Ye et al., 2015). However, until recently, hyperspectral acquisition technology was not mature enough to enable capture of temporal hyperspectral image sequences at reasonable frame rates. It is for this reason that, while there is an abundant literature on denoising (standard) video sequences or denoising still hyperspectral images, there are very limited contributions towards the denoising of hyperspectral image sequences.

Addressing the problem of denoising hyperspectral image sequences, we found that this preprocessing task poses a challenge due to several factors. Firstly, the addition of temporal information makes the data more complex to process. A preliminary solution to this would be to work separately with frames or process the temporal sequence band by band. However, this would not take advantage of temporal or inter-band information diversity. On the other hand, the type of noise that corrupts the image sequences may vary from one sequence to another and spectral variations that are a consequence of the temporal acquisition may appear. Thus, the denoising method should adapt to the particular characteristics of the hyperspectral scene, acquisition conditions

and noise model. Concerning operational issues, the availability of hyperspectral image sequences is extremely scarce, which makes the training and validation of denoising algorithms complex and cumbersome. Finally, taking under consideration that denoising is a preprocessing task, a desired characteristic of noise filtering methods is their real-time applicability.

In view of the above, this article presents a strategy for the denoising of hyperspectral image sequences based on the application of a cellular automata based approach called 4DCAF (4-dimensional cellular automata based filtering). The transition rules that govern the denoising behavior of 4DCAF are tuned following an evolutionary process called ECAF (evolutionary method for obtaining cellular automata filters), fed by synthetic hyperspectral image sequences functioning as training datasets. These are created reflecting similar characteristics to those of the real image sequences to be denoised. The proposed method presents a solution for the denoising of hyperspectral image sequences, simultaneously meeting the following aspects:

- The method is able to adapt to hyperspectral image sequences exhibiting different spatial, spectral and temporal features, as well as to different types of noise that may corrupt the image sequences. This is addressed by using appropriate synthetic image sequences in the training dataset for the evolutionary algorithm.
- The 4DCAF structure iteratively filters each frame of the sequence explicitly taking the temporal, spectral and spatial diversity into account.
- It presents intrinsically parallel low-level processing operations, which enables its implementation in a concurrent fashion over hardware such as GPUs.
- The proposed methodology outperforms several state-of-the-art algorithms on both simulated and real sequences.

The remainder of this article is organized as follows. Section 2 introduces the proposed 4DCAF structures. Sections 3 and 4 present the procedure fol-

lowed to determine the transition rules of the cellular automata based structures (ECAF), including the synthetic image sequence construction step. Section 5 addresses the experimental results of applying the denoising algorithms over both synthetic and real image sequences. Finally, some concluding remarks are summarized in section 6.

2. Cellular Automata based Filtering

A cellular automaton (Von Neumann et al., 1966) is a spatially extended decentralized model made up of cells which communicate only with their neighbors and continuously update their state. If those cells represent pixels and the neighbors are assigned based on spatial, spatio-temporal or spatio-spectro-temporal distance measurements, the automaton recreates the effect of an image filter. Cellular automata have been widely used for the study of complex systems based on multicomponent models, but they have been extended for image filtering only in some preliminary works (Hernandez and Herrmann, 1996; Popovici and Popovici, 2002; Wang et al., 2004a; Selvapeter and Hordijk, 2009).

In this context, we present a novel cellular automata based filter (CAF) architecturee, the 4-dimensional cellular automata based filter (4DCAF), for the denoising of hyperspectral image sequences. The essential purpose of the proposed cellular automaton structure is to modify the spectra of every cell so that after a number of iterations, every frame of the image sequence converges to a denoised version of it.

The rules that govern the behavior of CA-based filtering structures are automatically obtained through an evolutionary strategy called ECAF. Starting from a random rule set, the evolutionary process runs until it reaches a predefined performance and the resulting evolved cellular automaton, 4DCAF, constitutes the noise reduction filter. These rules will be optimized to denoise image sequences coming from specific sources and corrupted by particular types of noise depending on the dataset used to determine the fitness of the rule set during the evolutionary process. Fig. 1 shows a schematic representation of the process.

2.1. General operation of 4DCAF

In particular, in the case of a 4DCAF structure, a cell of the automaton is situated over each pixel of the image sequence and the state of the cell ($\boldsymbol{s_i}$) is given by an N-band spectrum, taking values for each band in the range $[0, 1]$. In fact, it is important to note here that 4DCAF always works with the whole spectrum at each point, without performing any kind of projection onto lower dimensionalities, with the aim of preserving all the information provided by the spectra.

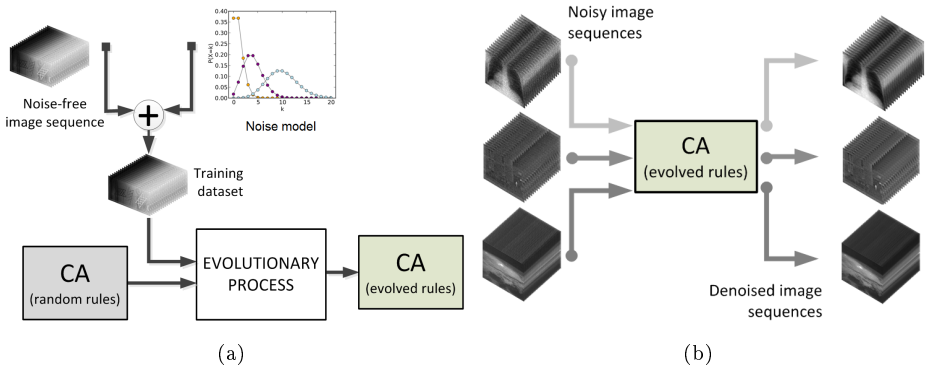
The 4DCAF structure is executed K times over a section of the hyperspectral image sequence centered on a temporal hyperspectral frame f, whose cells are gradually modified converging towards a denoised version of it. Once a frame is properly denoised, the CAF is moved to the next frame, performing again the denoising operation. The selection of K is based on a convergence analysis of the spectral values after each iteration in the application of the 4DCAF. The frame is considered denoised when the average spectral value difference between consecutive iterations is lower than a fixed threshold, ϵ.

The core of the operation of 4DCAF relies on the criteria which define the updating of the cell state. This updating is based on one hand, on information on the spectral values of the $N_S \times N_S \times N_T$ closest neighboring cells, where N_S and N_T are the spatial and temporal size of the window in which every cell is centered. On the other hand, it depends on the set of updating/transition rules that control the automaton behavior.

When dealing with hyperspectral images, in order to consider the spectral information of the neighboring cells in 4DCAF in a dimensionally independent manner, we have selected the spectral angle (SA), normalized between 0 and 1, as distance measure. Thus, for a cell i corresponding to a pixel spectrum, the normalized spectral angle, $\alpha_{i,j}$ with respect to its neighboring cell j is defined as:

$$\alpha_{i,j} = \frac{2}{\pi} cos^{-1}\left(\frac{\sum \boldsymbol{s_j} \cdot \boldsymbol{s_i}}{|\boldsymbol{s_j}||\boldsymbol{s_i}|}\right) \quad (1)$$

where the sum is performed over the components of the state of $\boldsymbol{s_i}$, i.e., the



(a) (b)

Figure 1: (a) Representation of the evolutionary process, ECAF, to automatically generate the 4DCAF rule set which will perform the denoising of image sequences with particular features and (b) representation of the application of the 4DCAF structures to real noisy image sequences

spectral dimension of a pixel.

2.2. Deciding on the appropriate transition rule and updating the state of the CA

In 4DCAF, local information is extracted based on the gradient vector ($\mathbf{G}$) of the cell, taking into account the pixels contained in an $N_S \times N_S \times N_T$ 3-dimensional window. Gradient measurements have been widely used in the image processing field, mainly to address common tasks such as border detection, segmentation, classification or denoising since these gradient values will provide information about the strength and direction of intensity changes in each pixel of the image sequence, which is usually very relevant. The crux of the process is that the CAF structure, through the transition rule set, is able to distinguish between fluctuations among neighboring pixels due to real edges or noise and to apply the proper modifications in order to reduce noise without losing sharpness.

In order to calculate the gradient vector, three different 3-dimensional masks (Fig. 2) are applied to the pixel, obtaining this way three gradient components: G_X, G_Y and G_T.

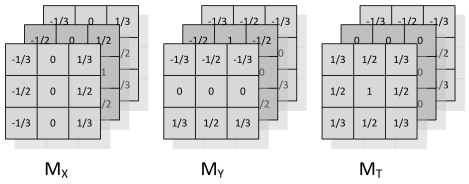


Figure 2: Gradient filter masks for $N_S = 3$ and $N_T = 3$. The value of every position of the filter mask is equal to $\frac{1}{d^2}$ where d^2 is the distance from that position to the pixel that is being filtered

The value of $\mathbf{G}$ when 4DCAF is applied to a hyperspectral image sequence is calculated as:

$$
\begin{aligned}
G_{X_{N_{S_i}}} &= \sum_{j=1}^{N_S \cdot N_S \cdot N_T} \alpha_{i,j} \cdot M_{X_{N_{S_j}}} \\
G_{Y_{N_{S_i}}} &= \sum_{j=1}^{N_S \cdot N_S \cdot N_T} \alpha_{i,j} \cdot M_{Y_{N_{S_j}}} \\
G_{T_{N_{T_i}}} &= \sum_{j=1}^{N_S \cdot N_S \cdot N_T} \alpha_{i,j} \cdot M_{T_{N_{T_j}}}
\end{aligned}
\tag{2}
$$

where $\alpha_{i,j}$ denotes the normalized spectral angle between a cell i and its neighboring cell j .

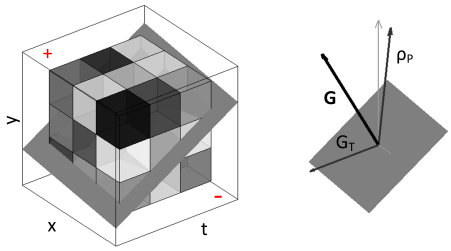


Figure 3: (Left) Example of spatio-temporal window when $N_S = 3$ and $N_T = 3$, including the plane defined by the gradient vector. Gray colored boxes represent spectral distances between neighboring pixels and the cell which is being evaluated. (Right) Gradient vector decomposition

The modulus of the spatial projection of the gradient vector ($\rho_P = \sqrt{G_X^2 + G_Y^2}$) and the absolute value of the temporal component (G_T) are taken

as the information the CAF needs in order to operate. Then, ρ_P is related to the spectral angle change in the spatial dimension, whereas G_T is associated to a spectral angle variation in the temporal dimension within a spatio-temporal $N_S \times N_S \times N_T$ 3-dimensional window. Additionally, a plane perpendicular to the gradient direction is defined based on this gradient vector in order to divide the neighborhood into two parts: a positive and a negative side (Fig. 3).

Thus, when the CA is applied to a particular cell, the **G** vector is obtained and the positive and negative sides are identified. Based on this separation, a transition rule will decide which pixels on each side are going to participate in the modification of the cell state and how they are going to contribute to this cell state updating process.

The CA is provided with a set of M transition rules and in every iteration of the CA structure, and for each pixel, one rule is selected to perform the updating.

The rules are made up of five parameters (2 state parameters + 3 updating parameters):

$$CA = \begin{Bmatrix} \rho_{r_{P_1}} & G_{r_{T_1}} & S_{r_{WS_1}} & T_{r_{WS_1}} & b_{r_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \rho_{r_{P_k}} & G_{r_{T_k}} & S_{r_{WS_k}} & T_{r_{WS_k}} & b_{r_k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \rho_{r_{P_M}} & G_{r_{T_M}} & S_{r_{WS_M}} & T_{r_{WS_M}} & b_{r_M} \end{Bmatrix} \tag{3}$$

$$\text{with} \begin{cases} \rho_{r_{P_k}} \in [0,1] \\ G_{r_{T_k}} \in [0,1] \\ S_{r_{WS_k}} \in \{1,3,5,\ldots,N_S\} \\ T_{r_{WS_k}} \in \{1,3,5,\ldots,T_S\} \\ b_{r_k} \in [0,1] \end{cases}$$

where $\rho_{r_{P_k}}$ and $G_{r_{T_k}}$ are the state parameters (they define which pixels will use this rule) and are related to the spatial and temporal projection of the

vector gradient; $S_{r_{WS_k}}$ and $T_{r_{WS_k}}$ denote the spatial and temporal size of the updating window which will define which pixels will be used to calculate the updated state of the current cell; b_{r_k} indicates the contribution of the pixels that are considered on the positive and negative side for the updating process. Subindex r is used here to avoid confusing rule parameters with the values of the gradient vectors extracted from the neighborhood information of a cell.

Table 1: Summary of 4DCAF

	Value	eq.
Type of temporal image sequence	hyperspectral	
Grid dimension	4D	
Cell state	$\mathbf{s_i}(\mathbb{R}^N)$	
Neighborhood information	$\alpha_{i,j}$	eq. 1
Spatio-spectro-temporal gradients	$G_{X_{N_{S_i}}} = \sum_{j=1}^{N_S N_S N_T} \alpha_{i,j} \cdot M_{X_{N_{S_j}}}$	eq. 2
	$G_{Y_{N_{S_i}}} = \sum_{j=1}^{N_S N_S N_T} \alpha_{i,j} \cdot M_{Y_{N_{S_j}}}$	eq. 2
	$G_{T_{N_{S_i}}} = \sum_{j=1}^{N_S N_S N_T} \alpha_{i,j} \cdot M_{T_{N_{S_j}}}$	eq. 2
Updating process	$\mathbf{s_{i,t+1}} = \bar{\mathbf{p}}_{\mathbf{i,t}} \cdot b_{r_q} + \bar{\mathbf{n}}_{\mathbf{i,t}} \cdot (1 - b_{r_q})$	(eq. 4)

To decide which rule, out of the rule set, is selected, a comparison between the neighborhood information (ρ_P and G_T) and the state parameters of each rule ($\rho_{r_{P_k}}$ and $G_{r_{T_k}}$) is carried out. As mentioned before, ρ_P and G_T, are obtained from projections of a gradient vector calculated within a 3-dimensional spatio-temporal $N_S \times N_S \times N_T$ window, being the particular values of N_S and N_T fixed and set manually. The automaton selects the rule whose first two parameters $\left(\{\rho_{r_{P_1}}, G_{r_{T_1}}\}, \{\rho_{r_{P_2}}, G_{r_{T_2}}\}, \ldots, \{\rho_{r_{P_M}}, G_{r_{T_M}}\}\right)$ are most similar in Euclidean terms to $\{\rho_P, G_T\}$. The selected rule, q, establishes the spatial and

temporal window size $\left(\{S_{r_{WS_q}}, T_{r_{WS_q}}\}\right)$ of the pixels that take part in the cell state updating process and the value of the updating parameter b_{r_q}.

The updating process of the new cell state for the 4DCAF structure is given by the following updating formula:

$$\mathbf{s_{i,t+1}} = \bar{\mathbf{p}}_{\mathbf{i,t}} \cdot b_{r_q} + \bar{\mathbf{n}}_{\mathbf{i,t}} \cdot (1 - b_{r_q}) \tag{4}$$

where $\mathbf{s_{i,t+1}}$ is the new spectrum of cell i and $\bar{\mathbf{p}}_{\mathbf{i,t}}$ and $\bar{\mathbf{n}}_{\mathbf{i,t}}$ denote the averaged spectra of all the pixels contained in the positive and negative sides of the spatio-temporal window defined by $\left\{S_{r_{WS_q}}, T_{r_{WS_q}}\right\}$.

This procedure is iteratively applied to all the cells of every frame producing, in the end, an updated and denoised hyperspectral image sequence —-provided that 4DCAF is properly tuned.

Table 1 presents the main characteristics of 4DCAF.

3. ECAF: Evolving the Cellular Automata

The efficiency of the proposed denoising method is subject to the set of transition rules that govern the behavior of the CAF structure. Setting the values of the parameters that make up the transition rules is a complex problem to solve manually. In order to obtain these rules automatically, we have chosen an optimization process in the form of an Evolutionary algorithm that encodes the CAF rule set (eq. 3). In particular, we have chosen a Genetic Algorithm (GA) in order to infer the set of transition rules.

3.1. Genetic Algorithm

Genetic Algorithms are basically search heuristics that mimic the process of natural evolution. Their basic principles were established by Holland (1975) and since then have been widely used to generate solutions to multiple and diverse optimization problems.

GAs operate over a population of individuals, data structures that encode the sets of transition rules that make up the CAF structures. Each generation, a new offspring population is created through the selection, based on the evaluation of individuals according to a cost/fitness function, and the variation (recombination and mutation) of individuals from the previous population.

The main components of the GA algorithm used here are:

- Representation: Each CAF structure that makes up the GA population is encoded as a vector consisting of $D = 5 \cdot M$ floating point values in the $[0, 1]$ interval, corresponding to M rules with five parameters each, two antecedents and three consequents for each rule.

- Population: NP D-dimensional parameter vectors are used as a population for each generation G. The initial vector population is chosen randomly and should cover the entire parameter space.

- Selection: In the selection step, a traditional *Stochastic Uniform* selection strategy was used to choose individual genomes from the population for later breeding. It is implemented as follows:

    1. The fitness function is scaled for each individual.
    2. We make each parent correspond to a section of a virtual line. The length of the section is proportional to its scaled value.
    3. A random step size is chosen.
    4. The algorithm moves along the line made up of all the contributions from the parents in steps of equal size. At each step, the algorithm allocates a parent from the section it lands on.

- Genetic operators: The next step is to generate a second generation of solutions from those selected through a combination of genetic operators: crossover (recombination) and mutation.

  The selected crossover function creates offspring by taking a weighted average of the parents. The weights associated to each parent are set by a single parameter, *Ratio*: $child = parent_1 + rand * Ratio * (parent_2 - parent_1)$.

  Mutation introduces small random changes in the individuals of the population to create mutated offspring. Mutation provides genetic diversity and enables the genetic algorithm to search a broader space. The selected mutation function generates directions that are adaptive with respect to the last successful or unsuccessful generation.

Concerning the evaluation of individuals, the fitness of a prospective CAF is determined by running it over noised synthetic training hyperspectral image sequences and comparing the denoising results to the desired original denoised images. The quality of the denoising obtained after applying the automaton is calculated using the mean squared error (MSE) between the original and denoised sequence.

Thus, the fitness function for a 4DCAF structure becomes:

$$MSE = \frac{1}{X \cdot Y \cdot N \cdot F} \sum_{f=1}^{F} \sum_{b=1}^{N} \sum_{x=1}^{X} \sum_{y=1}^{Y} [I(x, y, b, f) - I'(x, y, b, f)]^2 \qquad (5)$$

where $I(x, y, b, t)$ and $I'(x, y, b, t)$ represent the b-band spectral value of pixel (x, y) in frame f for the original and denoised sequences; X and Y are the spatial dimensions of the image, N is the number of bands and F is the number of frames of the sequences.

Every time a prospective CAF needs to be evaluated, it is run over a training image sequence and the denoising result it obtains is compared to the noise-free image sequence.

Thus, how the ECAF adapts CAF structures in order to perform a satisfactory denoising task will strongly depend on how the training image set is selected or created. The creation process of the training image dataset is described in more detail in the next section.

4. ECAF: Creating the training image dataset

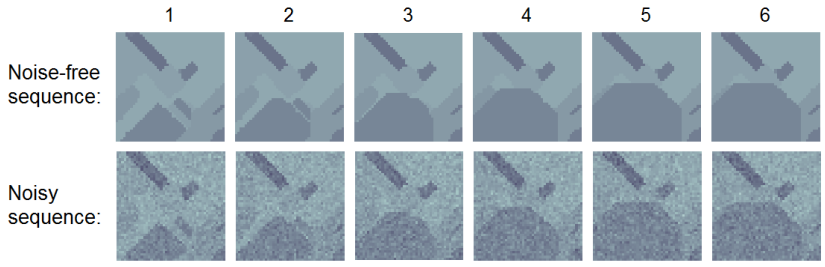
Evolving CA-based filtering structures capable of dealing with noise filtering effectively requires providing the evolutionary algorithm with suitable training image sequence datasets that should reflect the spatial, temporal and spectral properties of the type of real image sequences that require denoising, as well as particular characteristics of the type of noise that corrupts those real sequences.

The steps followed for the creation of synthetic hyperspectral image sequences for training are summarized below:

1. Creation of the noise-free image sequence (*reference sequence*)

   In order to obtain a noise-free synthetic hyperspectral image sequence for evolving 4DCAF structures, firstly, the ground truth of the first frame of the sequence is created. The ground truth of subsequent frames are a modified version of the first frame by growing or decreasing in size some selected regions. Then, global spectrum vectors are assigned to each area with the same associated label, thus obtaining the noise-free or reference multidimensional image sequence (Fig. 4).

2. Corruption of the reference sequence with noise (*noisy sequence*)
   Once the noise-free image sequence is created, some signal-independent and/or signal-dependent noise is added to it. Ideally, a previous study of the real sequences to be denoised is carried out in order to approximate which noise model is affecting them. Then, the synthetic reference image is noised following the type of noise model that has been determined.
   The following model can be used to approximate both additive signal-dependent and signal-independent random noise:



(a)

Figure 4: One band of noise-free and noisy synthetic hyperspectral image sequences used as training data set for the ECAF evolutionary strategy

$$\begin{aligned} g(x,y,b,t) &= f(x,y,b,t) + f(x,y,b,t)^{\gamma} \cdot u(x,y,b,t) + w(x,y,b,t) \\ &= f(x,y,b,t) + v(x,y,b,t) + w(x,y,b,t) \end{aligned} \tag{6}$$

In eq. 6, g denotes the signal value for each pixel including noise, f is the corresponding signal value without noise and γ is the exponential parameter of the model and the only parameter it contains. Based on the value of γ the model will represent different types of noise. Finally, u and w are zero-mean random variables with variances σ_u^2 and σ_w^2.

The most frequently considered signal dependent-noise models in denoising applications, due to their high presence in images, are film grain noise, multiplicative noise (speckle), and Poisson noise (also called shot or photon noise), (Kuan et al., 1985).

Namely, for $1/3 \leq \gamma \leq 1/2$ the model represents a typical configuration of the so-called film-grain noise, which can be observed on digital images and appears when images initially recorded on photographic film are digitized. This noise is therefore modeled using an additive model considering one signal dependent noise source and one signal independent noise source that are based on Gaussian probability functions.

Speckle noise is generated by coherent radiation as in the case of radars and synthetic aperture radars. A similar noise model is also present in ultrasounds, lasers and sonars. Speckle noise is described using a signal

independent noise term but by means of a multiplicative model where the signal is multiplied by this noise term ($\gamma = 1$ in eq. 6). As a result, the noise present in the corrupted image behaves as signal dependent noise and it is frequently modeled using zero mean additive noise with a variance obtained as a function of the uncorrupted signal at that point.

Finally, when an image system operates in low-light conditions a specific type of noise appears. It results from the discrete nature of the process of detection of the radiance, which is performed counting the number of incident photons on the sensor during a period of time for its later conversion into photoelectrons. This type of noise is called photon noise or shot noise and, since it is generated by the sum of independent random occurrences of discrete events, it behaves following a Poisson distribution. The exponent γ takes a value of 0.5 in eq. 6. This will generate the term $\sqrt{f} \cdot u$ that will approximate the Poisson distribution component. As reflected in the proposed approximation, this makes the level of noise dependent on the brightness of the scene, and more specifically, it leads to the magnitude of the noise term growing with the square root of the average intensity of the signal. Accordingly, it is in the case of low light conditions when photon noise becomes more significant provided that there are no other sources of noise (the one associated to the electronic amplifier of the signal produced by the CCD, for instance) with higher average intensity. Photon noise is relevant when dealing with sensors that work with different exposure levels or sensor gains. Poor lighting conditions appear in many and diverse image processing applications: night vision for dynamic environments, X-ray fluoroscopic captures when used for real-time medical interventions, etc. and they are associated to sensors which can be divided into three main categories: X-Ray images, Infrared Images and conventional images obtained in low light conditions. This type of noise is very relevant here due to the way most hyperspectrometers operate. They usually split the light corresponding to one pixel into as many portions as bands are to be detected by the sensor and direct each portion to one sensing element

in the detector. Thus, as we are usually talking about more than a hundred bands, light intensity reaching each sensing element in the detector is usually quite low.

5. Application of 4DCAF to image sequences

This section is devoted to the presentation of the experimental results obtained from applying the 4DCAF structure and to its comparison to recent successful methods from the state of the art. This comparison has been carried out using both synthetic and real hyperspectral image sequences. All the state-of-the-art methods cited in section 1 have been tested over the hyperspectral sequences considered in this work. From all the results, only those that provided better performances are shown here. Thus, the methods selected for comparisons are MTSNMF (Ye et al., 2015), K-SVD (Aharon et al., 2006), BM3D (Dabov et al., 2007), BM4D (Maggioni et al., 2013) and DNTDL (Peng et al., 2014), respectively.

K-SVD and BM3D treat each frame and spectral band separately. This has been called frame-wise band-wise (fw-bw) processing, whereas MTSNMF, BM4D and DNTDL are applied frame-wise (fw).

Table 2: Genetic Algorithm (GA) and CAF parameters

Name	Value
(Genetic Algorithm)	
Number of parameters:	75 ($15\ rules \times 5\ parameters/rule$)
NP (Population Size):	200
Elite count (number of individuals that are guaranteed to survive to the next generation):	$0.05 \times NP$
CR (Fraction of the next generation, other than elite children, that are produced by crossover):	0.8
Ratio (Crossover):	0.7
Stopping criterion:	Max. number of generations OR Fitness function tolerance
(CAF)	
N_S	7
N_T	7

5.1. Validation of 4DCAF over synthetic hyperspectral image sequences

This subsection addresses the quantitative validation of the 4DCAF method by applying the CAF structure to a synthetic hyperspectral image sequence.

Before the application of the 4DCAF structure over the synthetic hyperspectral image sequence, the set of transition rules that define its behavior have been evolved using the ECAF evolutionary method, feeding the evaluation step of the genetic algorithm with synthetically created training datasets.

On the other hand, the synthetic image sequence created for validation purposes will be noised following the same estimated noise model exhibited by the real test image sequences shown in subsection 5.2. Thus, before corrupting both the training dataset employed in the ECAF method and the validation synthetic hyperspectral image, it will be necessary to estimate the noise model of the real hyperspectral image sequence referred in subsection 5.2.

The analysis of the type of noise model that affects the real sequence has been carried out by representing the variance (σ) versus the averaged intensity value (μ) for each spectral band of the sequence (Fig. 5). Both σ and μ values were

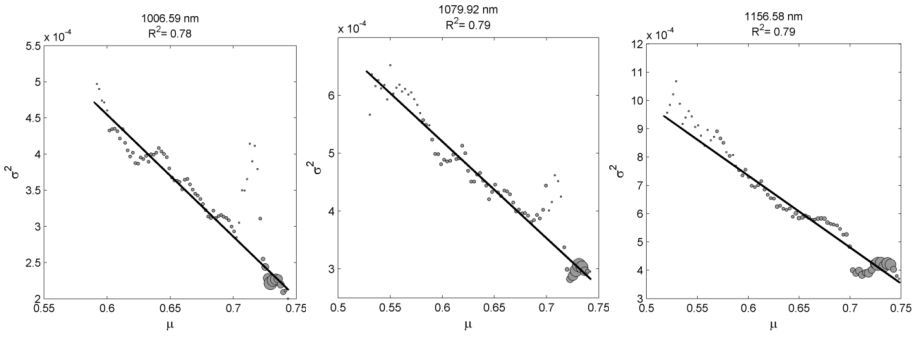
obtained considering values in a spatial window of 3×3 pixels and a temporal window of 25 frames for regions in the image sequence without moving objects. The radius of the circles in these graphs represent the density of pixels in each range of intensity that presents a certain noise variance value. Fig. 5 shows the noise analysis for three selected spectral bands. The value of the linear regression coefficient, R, is displayed in the title of each graph presenting a good fit to the linear models.

In order to construct each training image sequence used as input for the evolutionary algorithm, firstly, a ground truth map sequence was created. Then, a real spectral signature was associated to each labeled region of the ground truth map sequence, thus creating the reference or noise-free hyperspectral image sequence. After that, each band of the reference sequence was corrupted using the following model:

$$g(x, y, b, t) = f(x, y, b, t) + (p_2(b) + p_1(b) \cdot f(x, y, b, t))^{\frac{1}{2}} \cdot u(x, y, b, t) \quad (7)$$

where g denotes the spectral value being noised, f the corresponding signal value without noise; u is a zero-mean random variable with standard deviation 1; and p_2 and p_1 are the offset and the slope of the linear regression fit for each band b, respectively. In total, three different synthetic sequences have been generated to form the training dataset.

Regarding the synthetic image sequence used for validating the 4DCAF structure, the creation process differs from the training dataset in the selection of the spectral signatures for creating the noise-free hyperspectral image sequence. The spectral signatures (composed of 129 bands) have been extracted from the real sequence presented in subsection 5.2 using the following procedure: firstly, N pixel spectra have been selected from the real sequence aiming to be as spectrally different as possible from one another (in terms of spectral angle), where N is the number of different labels found in the ground truth image sequence; then, these N spectra have been filtered achieving a smoothed effect. The reference image sequence has been corrupted following the same procedure as indicated for the construction of the training hyperspectral image sequence



(a) (b) (c)

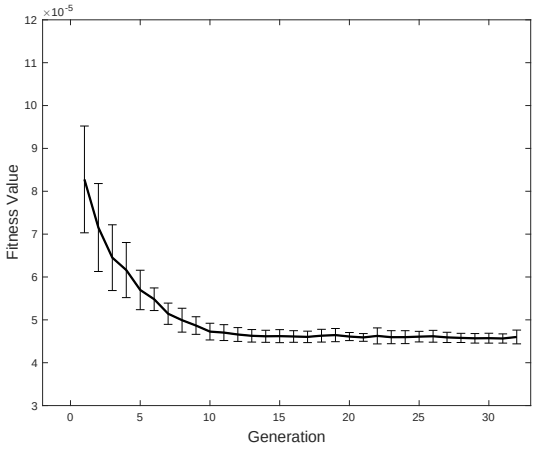
Figure 5: Noise variance vs. intensity signal value graph for the 1006.59 nm (a), 1079.92 nm (b) and 1156.58 nm (c) bands of the real hyperspectral image shown in Fig. 11

dataset.

The technical specifics of the ECAF procedure and 4DCAF structures used are summarized in table 2 and captures of the training datasets are shown in Fig. 4. Fig. 6 displays the average performance of the genetic algorithm over time. This graph represents the average value of the mean fitness of the population after executing the genetic algorithm 6 times. The standard deviation of the mean fitness is also represented using error bars. The decreasing trend of the error followed by the curve is clear, converging after about 10 generations.

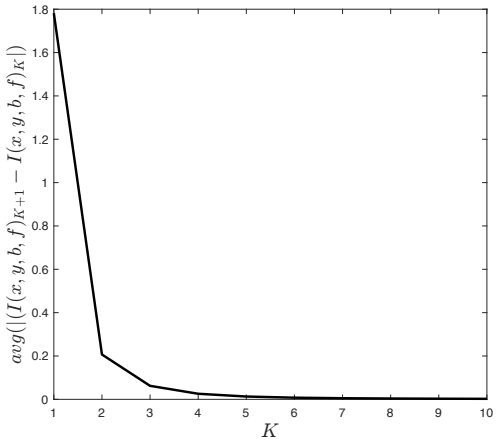
The 4DCAF structure evolved using the mentioned training dataset and the evolutionary parameters of table 2 will represent the multidimensional CA-based filter used to denoise both the synthetic sequence of this subsection and the real sequences of the next subsection. The 4DCAF has been applied 5 times over each frame of the real hyperspectral sequence based on a previous analysis of convergence (Fig. 7): the 4DCAF is applied until the average spectral value difference between consecutive iterations is lower than fixed threshold, ϵ, in this case, $\epsilon = 0.001$.

Turning now to the evaluation of the performances of the algorithms, we have obtained four quantitative picture quality indices (PQI), including peak signal-to-noise ratio (PSNR), structure similarity (SSIM) (Wang et al., 2004b),



(a)

Figure 6: Average performance of the genetic algorithm over time



(a)

Figure 7: Average spectral value difference between consecutive iterations in the application of the 4DCAF

feature similarity (FSIM) (Zhang et al., 2011) and spectral angle mapper (SAM) (Yuhas et al., 1993).

The 4DCAF structure was applied directly to the synthetic sequence, while in the case of the other methods, before the application of fw-MTSNMF, fw-bw-KSVD, fw-bw-BM3D, fw-DNTD and fw-BM4D, the hyperspectral sequence was transformed using a variance-stabilizing transformation (VTS) (Starck et al., 1998). A summary of the parameters used for the denoising are shown in Table 4.

Quantitative and visual results are presented in Table 3 and Fig. 8 respectively. The picture quality indices demonstrate that the proposed method is able to efficiently operate over noisy hyperspectral image sequences, consistently providing values for the indices that are much better than those given by the other methods. On the other hand, the spectral bands from a selected frame of the sequence represented in Fig. 8 show that 4DCAF reduces the noise and preserves small structures and sharp edges.

Fig. 9 depicts the absolute deviation between the reference and the denoised spectral images of a selected frame from the synthetic hyperspectral image sequence shown in Fig. 8. It can be observed that fw-bw-KSVD and fw-BM4D introduce spot-type noise. Moreover, the denoising quality provided by the fw-BM4D method is band-dependent, as it can clearly be noticed for the 853.3 nm spectral band. The fw-bw-BM3D method produces a blurring effect on the edges while at the same time presenting two-dimensional noise fluctuations. The behavior in flat areas is quite good for the fw-DNTDL denoising result but heavy speckle-type noise is present close to the borders. The fw-MTSNMF method preserves the sharp boundaries well but reveals noisier flat areas. Finally, the 4DCAF method provides the best result in terms of flat areas, low deviation and preservation of sharpness close to the edges.

The graphs in Fig. 10 show the per-frame and per-band PSNR, SSIM and FSIM quality indices of the proposed 4DCAF, fw-MTSNMF, fw-bw-KSVD, fw-bw-BM3D, fw-DNTDL and fw-BMd4D methods applied to the synthetic hyperspectral image sequence under analysis. Regarding the per-frame represen-

tation, 4DCAF outperforms the other five methods after the first two frames of the sequence in terms of PSNR, SSIM and FSIM. The loss of quality in the first frames is due to the lack of temporal information associated with the initial steps of the process. On the other hand, observing the per-band representation, it can be noticed that the denoising performances vary from band to band for all the methods, exhibiting the 4DCAF method the best quality values for all the spectral bands. Furthermore, considering the SSIM and FSIM indices, the fluctuations between bands are much lower for the 4DCAF result.

Table 3: PQI comparison of the proposed 4DCAF, fw-bw-KSVD,fw-bw-BM3D, fw-DNTDL, MTSNMF and fw-BM4D applied to a synthetic image sequence corrupted by signal dependent noise

Method	PSNR	SSIM	FSIM	SAM
4DCAF	**49.62**	**0.993**	**0.982**	**0.003**
fw-MTSNMF (Ye et al., 2015)	46.02	0.986	0.957	0.007
fw-bw-KSVD (Aharon et al., 2006)	41.36	0.972	0.932	0.010
fw-bw-BM3D (Dabov et al., 2007)	42.85	0.976	0.924	0.009
fw-DNTDL (Peng et al., 2014)	44.99	0.983	0.971	0.006
fw-BM4D (Maggioni et al., 2013)	44.01	0.984	0.958	0.010

5.2. Application of 4DCAF to real hyperspectral image sequences

Once the proposed method has been validated over synthetic sequences, we have applied 4DCAF to two different real hyperspectral image sequences of dispersion of chemical plumes acquired by a LWIR sensor. The sequences consist of 25 and 31 frames respectively, each one containing 129 spectral bands in the $[853, 1.280]$ nm range. For these sequences, since there is no noise free version available, a first evaluation of the denoising has to be performed visually. Fig. 11 and 12 demonstrate that the 4DCAF method produces the most satisfactory result: the background appears more homogeneous, whereas the shape of the plumes look sharper than in the case of the other methods.

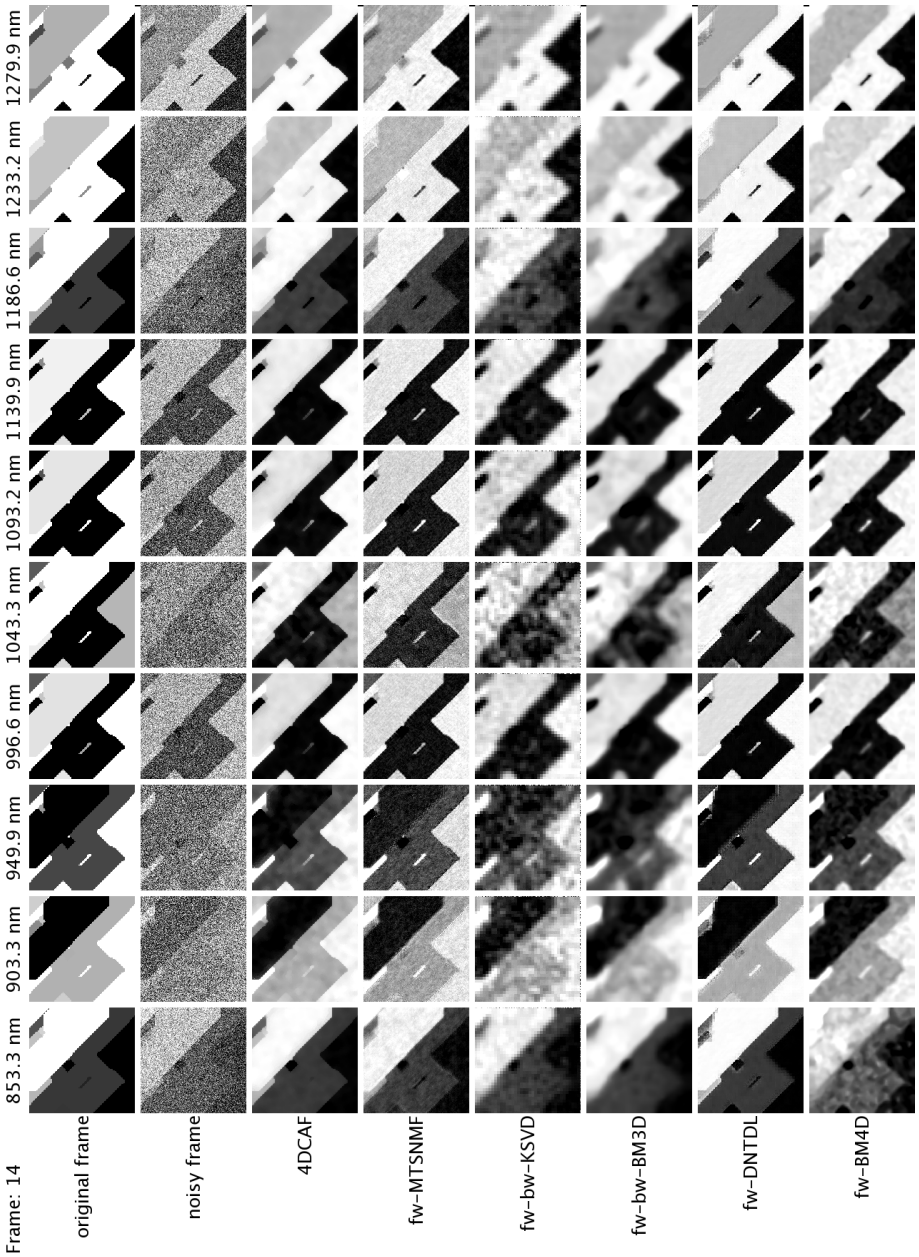


Figure 8: Original, noisy and denoised images of a selected frame from a synthetic hyperspectral image sequence. Only ten bands (out of 129) are shown

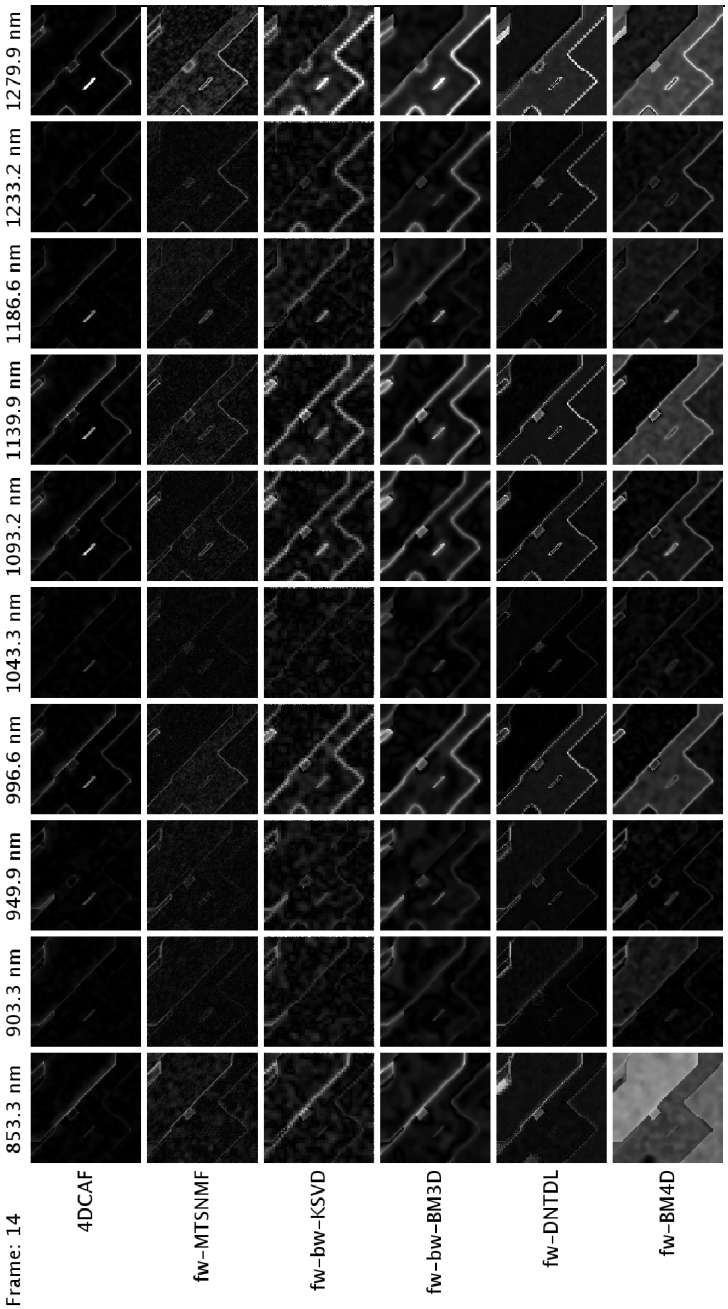


Figure 9: Absolute deviation between the reference and the denoised images of a selected frame from the synthetic hyperspectral image sequence shown in Fig. 8. Only ten bands (out of 129) are shown

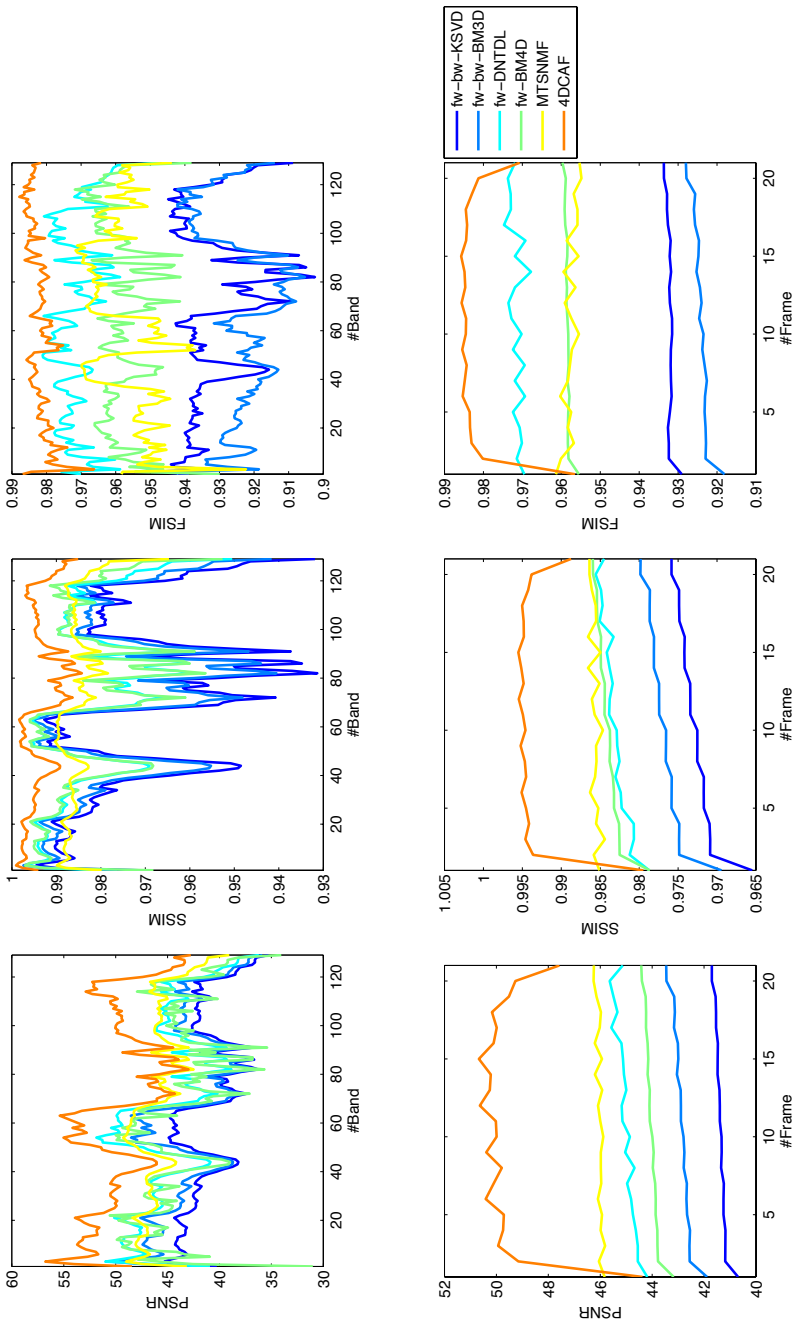


Figure 10: Per-frame and per-band PSNR, SSIM and FSIM comparison of the proposed 4DCAF, fw-bw-KSVD, fw-bw-BM3D,fw-DNTDL, MTSNMF and fw-BMd4D methods applied to the synthetic hyperspectral image sequence shown in Fig. 8

Also for assessing the denoising of real hyperspectral image sequences, we have considered a 2D representation (Fig. 13b and 14b) which plot the fluctuations in terms of spectral angle between contiguous pixels belonging to a string of pixels selected following a spatio-temporal zigzag path (marked with red circles in the left image of Fig. 13a and 14a) which temporally and spatially traverses the sequences. Furthermore, Fig. 13c and 14c plot the same values presented in Fig. 13b and 14b but sorted in ascending order for the X-axis according to the minimum spectral angle fluctuation selected from the noisy and denoised pixels. Both 2D representations reveal that the variations between contiguous pixels, in terms of spectral angle, are lower when the 4DCAF structure is applied as compared to the other denoising methods under consideration.

6. Conclusions

This work has presented a strategy for the denoising of hyperspectral image sequences. The proposed method deals with some issues associated to the denoising problem: the lack of available noised and noise-free hyperspectral image sequences to validate and train algorithms; the adaptability of methods to different types of noise that may corrupt the sequences; and the joint processing of temporal, spatial and spectral information, taking into account the inter-dimensional diversity.

In order to solve these problems, a methodology based on a CA-based filtering structure (called 4DCAF) and an evolutionary approach (ECAF), was proposed. The adaptation of the CAF to properly perform the denoising considering a particular type of noise model is carried out by an evolutionary technique after appropriately selecting synthetic noisy image sequences, which are used to evaluate the individuals of the population. Therefore, an advantage of the ECAF method is its versatility for adapting the rule set of the CAF structure to the noise model exhibited by the image source in an automatic manner by just by modifying the training image sequences.

The 4DCAF structure has been validated over synthetic and real hyper-

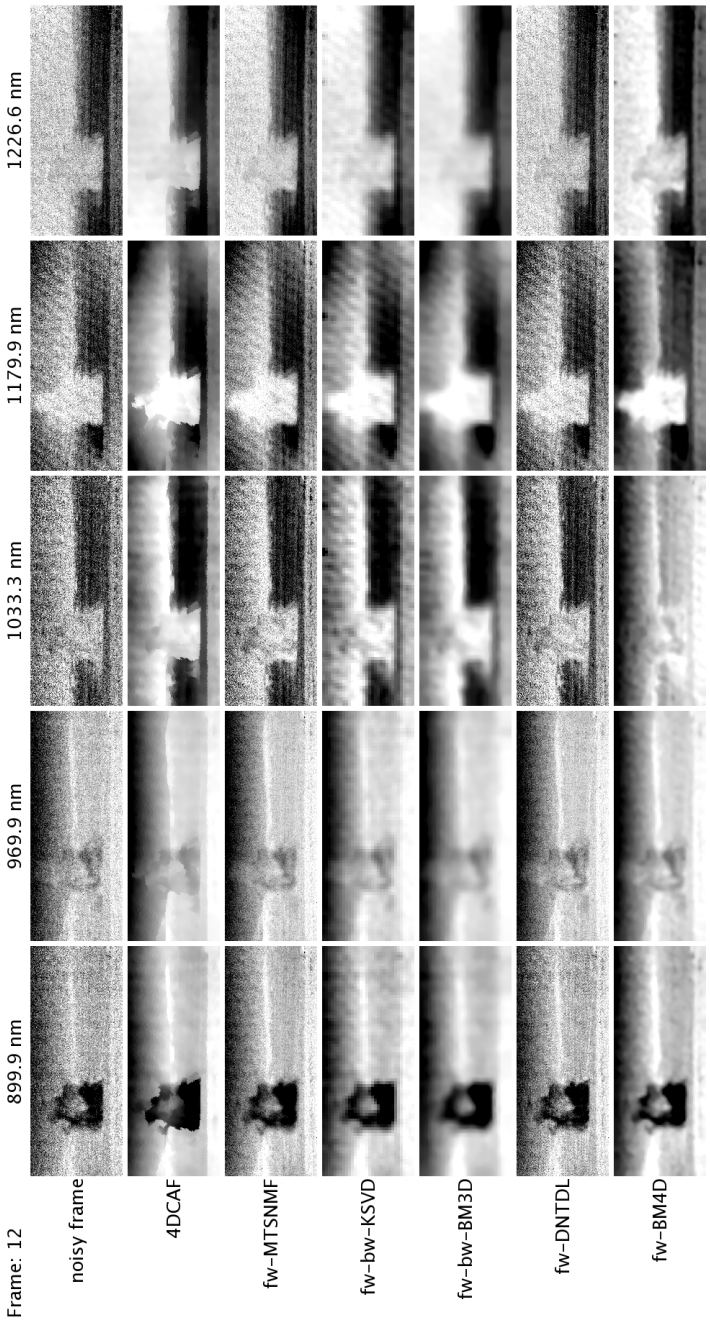


Figure 11: Noisy and denoised images of a selected frame from a first real hyperspectral image sequence representing a chemical plume. Only five bands (out of 129) are shown

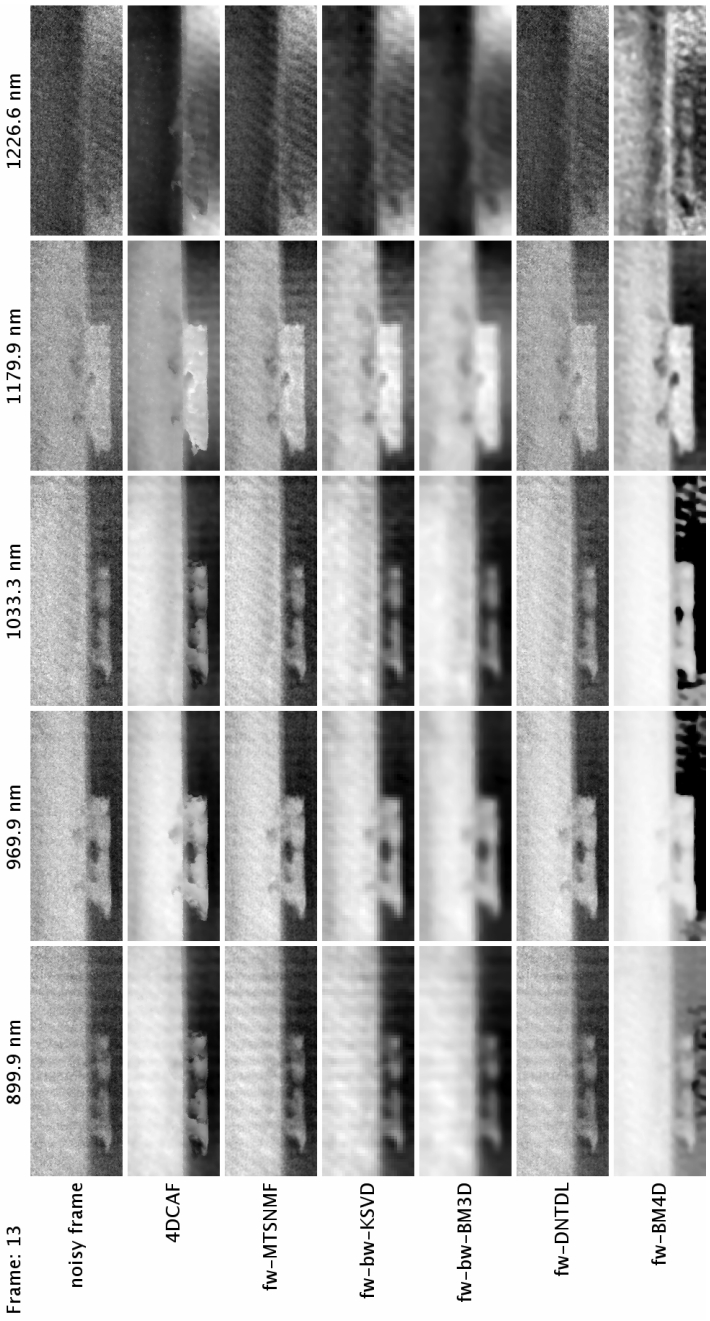
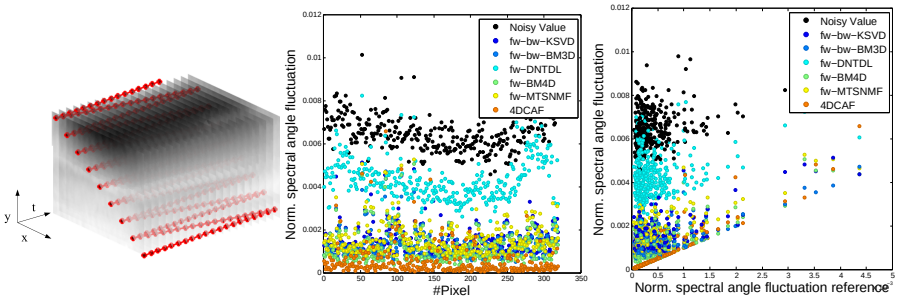
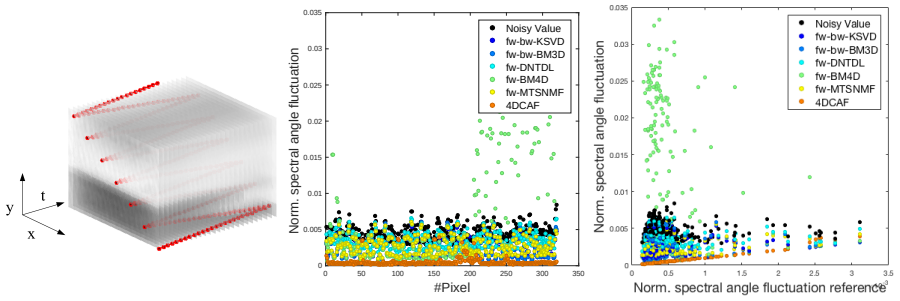


Figure 12: Noisy and denoised images of a selected frame from a second real hyperspectral image sequence representing a chemical plume. Only five bands (out of 129) are shown



(a) (b) (c)

Figure 13: (a)Temporal sequence of Fig. (only one band is shown) with the selected pixels represented by red circles; (b) Intensity fluctuation (spectral angle distances) from noisy and denoised sequences (Fig. 11) between contiguous pixels selected following the spatio-temporal zigzag path marked with red circles; (c) Same values as presented in (b) but sorted in ascending order for the X-axis according to the minimum spectral angle fluctuation selected from the noisy and denoised values



(a) (b) (c)

Figure 14: (a)Temporal sequence of Fig. (only one band is shown) with the selected pixels represented by red circles; (b) Intensity fluctuation (spectral angle distances) from noisy and denoised sequences (Fig. 12) between contiguous pixels selected following the spatio-temporal zigzag path marked with red circles; (c) Same values as presented in (b) but sorted in ascending order for the X-axis according to the minimum spectral angle fluctuation selected from the noisy and denoised values

spectral image sequences. To that end, an analysis of the noise model that affects the real sequence has been performed. The estimated noise model has been used to corrupt synthetic images created for the validation of the method. After the application of the evolved 4DCAF structure, the performances are excellent when compared to alternative state of the art filtering strategies, both from a quantitative perspective, as observed on synthetic sequences, and from a qualitative point of view using visual analysis over real hyperspectral sequences featuring the release and dispersion of gas plumes following an explosion.

In addition to the successful performance of the approach, an important perspective of this work is related to its potential for efficient parallel implementations. Since the proposed method is based on cellular automata, which are intrinsically distributed structures, it can be easily implemented by distributing it over high speed GPU based platforms, thus significantly increasing its suitability for real time applications.

Table 4: Summary of parameters used for the denoising methods considered here

Method	Parameters
fw-bw KSVD	Block Size = [8, 8]
	Number of training signals = 200
	Step Size = [4, 4]
	Dictionary Size = 128
fw-bw BM3D	Quality/complexity trade-off profile selection = Normal
fw-DNTDL	Block Size = [8 8]
	Size of overlaps = [4 4]
	Number of clusters in block matching = 25
fw-BM4D	Quality/complexity trade-off profile selection = Normal
fw-MTSNMF	Patch Size = [7,7]
	Overlap Pixel = 2
	Dictionary Size = 4802

Acknowledgements

We thank Andrea Bertozzi, Department of Mathematics, University of California - Los Angeles (UCLA), for her collaboration on the dataset provided by the US Defense Threat Reduction Agency and the National Science Foundation through NSF grant DMS- 1118971.

This work was partially supported by the Ministry of Science and Innovation, Government of Spain, co-funded by the FEDER funds of the European Union, under contract TIN2015-63646-C5-1-R, and by Xunta de Galicia and redTEIC network (ED341D R2016/012)

This work was also partially funded through DGA, under grant 2015-60-001200.470.75.01.

References

References

Aharon, M., Elad, M., Bruckstein, A., 2006. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing 54, 4311–4322.

Buades, A., Coll, B., Morel, J.M., 2005. A non-local algorithm for image denoising, in: Computer Vision and Pattern Recognition, 2005 IEEE Computer Society Conference on, IEEE. pp. 60–65.

Dabov, K., Foi, A., Egiazarian, K., 2007. Video denoising by sparse 3d transform-domain collaborative filtering, in: Proc. 15th European Signal Processing Conference, p. 7.

Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K., 2006. Image denoising with block-matching and 3d filtering, in: Electronic Imaging 2006, International Society for Optics and Photonics. pp. 606414–606414.

Hernandez, G., Herrmann, H.J., 1996. Cellular automata for elementary image enhancement. Graphical Models and Image Processing 58, 82–89.

Holland, J.H., 1975. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press.

Kuan, D.T., Sawchuk, A., Strand, T.C., Chavel, P., et al., 1985. Adaptive noise smoothing filter for images with signal-dependent noise. IEEE Transactions on Pattern Analysis and Machine Intelligence , 165–177.

Lam, A., Sato, I., Sato, Y., 2012. Denoising hyperspectral images using spectral domain statistics, in: Pattern Recognition, 2012 21st International Conference on, IEEE. pp. 477–480.

Liao, C.S., Choi, J.H., Zhang, D., Chan, S.H., Cheng, J.X., 2015. Denoising stimulated raman spectroscopic images by total variation minimization. The Journal of Physical Chemistry C 119, 19397–19403.

Liu, X., Bourennane, S., Fossati, C., 2012. Denoising of hyperspectral images using the parafac model and statistical performance analysis. IEEE Transactions on Geoscience and Remote Sensing 50, 3717–3724.

Luisier, F., Blu, T., 2008. Sure-let multichannel image denoising: interscale orthonormal wavelet thresholding. IEEE Transactions on Image Processing 17, 482–492.

Maggioni, M., Katkovnik, V., Egiazarian, K., Foi, A., 2013. Nonlocal transform-domain filter for volumetric data denoising and reconstruction. IEEE Transactions on Image Processing 22, 119–133.

Peng, Y., Meng, D., Xu, Z., Gao, C., Yang, Y., Zhang, B., 2014. Decomposable nonlocal tensor dictionary learning for multispectral image denoising, in: Computer Vision and Pattern Recognition, 2014 IEEE Conference on, IEEE. pp. 2949–2956.

Popovici, A., Popovici, D., 2002. Cellular automata in image processing, in: Fifteenth International Symposium on Mathematical Theory of Networks and Systems.

Portilla, J., Strela, V., Wainwright, M.J., Simoncelli, E.P., 2003. Image denoising using scale mixtures of gaussians in the wavelet domain. IEEE Transactions on Image Processing 12, 1338–1351.

Priego, B., Veganzones, M.A., Chanussot, J., Amiot, C., Prieto, A., Duro, R.J., 2013. Spatio-temporal cellular automata-based filtering for image sequence denoising: Application to fluoroscopic sequences, in: Image Processing, 2013 20th IEEE International Conference on, IEEE. pp. 548–552.

Renard, N., Bourennane, S., Blanc-Talon, J., 2008. Denoising and dimensionality reduction using multilinear tools for hyperspectral images. Geoscience and Remote Sensing Letters 5, 138–142.

Rudin, L.I., Osher, S., Fatemi, E., 1992. Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena 60, 259–268.

Salmon, J., Harmany, Z., Deledalle, C.A., Willett, R., 2014. Poisson noise reduction with non-local pca. Journal of Mathematical Imaging and Vision 48, 279–294.

Selvapeter, J.P., Hordijk, W., 2009. Cellular automata for image noise filtering, in: Nature & Biologically Inspired Computing, 2009 World Congress on, IEEE. pp. 193–197.

Starck, J.L., Murtagh, F.D., Bijaoui, A., 1998. Image processing and data analysis: the multiscale approach. Cambridge University Press.

Von Neumann, J., Burks, A.W., et al., 1966. Theory of self-reproducing automata. IEEE Transactions on Neural Networks 5, 3–14.

Wang, H.m., Guo, S.d., Yu, D.h., 2004a. A new ca method for image processing based on morphology and coordinate logic. Application Research of Computers 1, 081.

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004b. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 13, 600–612.

Ye, M., Qian, Y., Zhou, J., 2015. Multitask sparse nonnegative matrix factorization for joint spectral–spatial hyperspectral imagery denoising. IEEE Transactions on Geoscience and Remote Sensing 53, 2621–2639.

Yuhas, R.H., Boardman, J.W., Goetz, A.F., 1993. Determination of semi-arid landscape endmembers and seasonal trends using convex geometry spectral unmixing techniques. Summaries of the 4th Annual JPL Airborne Geoscience Workshop .

Zhang, L., Zhang, L., Mou, X., Zhang, D., 2011. Fsim: a feature similarity index for image quality assessment. IEEE Transactions on Image Processing 20, 2378–2386.