Hyperspectral Image Denoising using Dictionary Learning
Cassio F. Dantas, Jérémy E Cohen, Rémi Gribonval

To cite this version:

HAL Id: hal-02175630
https://inria.hal.science/hal-02175630
Submitted on 5 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
HYPERSPECTRAL IMAGE DENOISING USING DICTIONARY LEARNING

Cássio F. Dantas, Jérémy E. Cohen, Rémi Gribonval
Univ Rennes, Inria, CNRS, IRISA
Rennes, France

ABSTRACT
Hyperspectral images are corrupted by noise during their acquisition. In this work, we propose to efficiently denoise hyperspectral images under two assumptions: (i) noiseless hyperspectral images in matrix form are low-rank, and (ii) image patches are sparse in a proper representation domain defined through a dictionary. These two assumptions have already led to state-of-the-art denoising methods using fixed Wavelet transforms. We propose to rather learn the dictionary from hyperspectral images, a task commonly known as dictionary learning. We show that the dictionary learning approach is more efficient to denoise hyperspectral images than state-of-the-art methods with fixed dictionaries, at the cost of a larger computation time.

Index Terms— Hyperspectral image, denoising, sparsity, low-rank, dictionary learning.

1. INTRODUCTION
Hyperspectral imaging has become a major image modality over the last years, largely thanks to the development of spectral sensors. Hyperspectral images collect reflectance spectra with significant spectral resolution (~200 wavelengths) for a large number of pixels (~1000 × 1000 pixels). These images contain a lot of information regarding the composition of the scene, and can therefore be used in remote sensing for monitoring forest or coastal regions, or in chemometrics to study the composition of chemical compounds [1].

Removing noise in HSI is an important pre-processing step for any learning task such as segmentation, detection or spectral unmixing, and has therefore been studied extensively in the literature. State-of-the-art techniques exploit two key properties [2]: HSI typically are low rank matrices once vectorized in the spatial dimensions; and HSI (or small patches of it) are sparse in some well chosen bases such as wavelets. In particular, the state-of-the-art method HyRes [3] makes use solely of these two assumptions to very efficiently denoise remotely acquired HSI with minimal computation time.

In this work, we propose to push further the boundaries of HSI denoising performance by switching from a fixed dictionary setting (wavelets) to a learned dictionary setting. Dictionary Learning (DL) is well established in HSI applications such as unmixing [4] and classification [5]. It has also been previously applied to HSI denoising in several occasions (see [6] and references therein) but rarely coupled with a low-rank decomposition. Although the low-rankness of the HSI is enforced via nuclear norm in [7], the dimensionality reduction promoted by an explicit low-rank factorization is not exploited at all as the full-size HSI is manipulated. Moreover, in [7], like in most DL methods for traditional image denoising [8], the noise variance is supposed known, while here we use an efficient heuristic to estimate it.

In what follows, we first formalize the denoising task and the low-rank and image sparsity assumptions in Section 2. In Section 3 we present our generic DL-based algorithm for HSI, and instantiate it with two DL algorithms, K-SVD [9] and SuKro [10]. Finally, in Section 4 we show on artificially noised HSIs that the proposed approach outperforms state-of-the-art methods for HSI denoising at the cost of increased computation time that is reasonable for offline applications.

2. SPARSE AND LOW-RANK MODELING
A hyperspectral image is composed of two spatial dimensions (image space) and one spectral dimension, and thus naturally represented as a 3 dimensional cube of data, say $H \in \mathbb{R}^{n_1 \times n_2 \times p}$, where $n_1$ and $n_2$ correspond to the spatial dimensions and $p$ is the number of spectral bands.

Nevertheless, a hyperspectral image is also often represented in its matrix (2 dimensional) form by vectorizing its spatial dimensions at each spectral band. In an additive noise
model, this leads to:

\[ H = X + N \]  

(1)

where \( H \in \mathbb{R}^{n \times p} \) (with \( n = n_1 n_2 \)) containing as its \( j \)-th column the vectorized observed image at band \( j \). \( X \) and \( N \) are respectively the noise-free unknown signal, and the noise matrix, both \((n \times p)\) matrices.

### 2.1. Low-rank assumption

An HSI \( X \) is commonly modeled as a low-rank matrix in the literature \([3,11]\). The classical linear mixture model \([12]\) is itself a low-rank model \( X = AS^T \), in which the image is decomposed into a few subregions called source images (as the columns of \( A \) each containing a certain material with distinct spectral signature (in the columns of \( S \)). The HSI rank \( r \) in this case is the number of materials present in the image.

The singular value decomposition (SVD) is a way of obtaining an analogous low-rank decomposition of \( X \), even though it does not, in principle, promote spectral unmixing. It leads to the noisy low-rank model

\[ H = U \Sigma V^T + N \]  

(2)

where the columns of \( U, V \in \mathbb{R}^{n \times r} \) respectively a set of vectorized \textit{eigen-images} and the associated spectral components, and \( \Sigma \in \mathbb{R}^{r \times r} \) is a diagonal singular value matrix. Figure 1 illustrates the described models and notations.

### 2.2. Sparsity assumption

The eigen-images \( U_k = \text{unvec}(u_k) \) (with \( u_k \) the \( k \)-th column of \( U \)) can, then, be sparsely represented in a well-suited base—or, more generally, representation domain. In the noisy case, sparse reconstruction thus works as a form of denoising. Indeed, sparse modeling for images has been used for decades now \([8,13]\), with very good performance in denoising tasks.

State-of-the-art techniques in Hyperspectral image restoration \([3]\) use a 2-D (Kronecker) orthogonal wavelet basis \( B \) in which the vectorized eigen-images \( u_k \) are represented as a sparse set of wavelet coefficients \( w_k \) (\( k \)-th column of sparse coefficient matrix \( W \)): \( U = BW \). The full model becomes:

\[ H = BW \Sigma V^T + N \]  

(3)

### 2.3. A typical processing pipeline

In view of model \((3)\), a usual denoising approach given the corrupted signal \( H \)—which is no longer low-rank due to the noise—consists in estimating \( \hat{U}, \hat{\Sigma} \) and \( \hat{V} \) via SVD truncation and further denoising \( \hat{U} \) via 2D-Wavelet shrinkage. The spectral component of the low-rank model \( \hat{V} \) can also be denoised (for instance, \([14]\) uses a 1D-Wavelet shrinkage).

### 3. PROPOSED APPROACH

A first part of our contribution consists in replacing the classic orthogonal Wavelet basis in the described pipeline by a dictionary both 1) overcomplete and 2) learned from data.

#### 3.1. Overcomplete learned dictionary

Dictionary learning is a consolidated research topic \([8,10]\). Both the overcompleteness and the fact of learning a dictionary from a targeted type of data has been shown to improve the performance of natural image denoising tasks \([8]\) when compared to the classical orthogonal Wavelet-based approach. Any existing dictionary learning approach could be used for our purpose here. We propose to use two methods:

1. K-SVD \([9]\) as a standard (baseline) algorithm for learning unstructured dictionaries;

2. SuKro \([10]\) as an algorithm for learning Kronecker-structured dictionaries. This type of structure arises naturally when dealing with vectorized 2D-signals (the eigen-images in the columns of \( U \)). The usual Wavelet approaches (e.g. in the state-of-the-art HyRes \([3]\)) are themselves Kronecker-structured.

An important drawback of the overcompleteness (and consequently, non-orthogonality) of the dictionary can be pointed out: it significantly complicates the estimation of the sparse coefficients when compared to the orthogonal case (for instance, in HyRes \([3]\), this step comes down to a simple thresholding operation). Here, we will need to resort to the so-called sparse coding algorithms \([13]\) which are a sub-optimal way of solving this—now ill-posed—problem.

Consequently, computational complexity (and running time) is significantly bigger for the proposed method when compared to the literature. However, we will see in Section \([4]\) that a performance gain is obtained. Therefore, except in very constrained cases, the proposed technique is worthy.

#### 3.2. A patch-based denoising approach

In order to avoid having a prohibitively large dictionary \( D \in \mathbb{R}^{n \times m} \) (with \( m > n \)), which would be very hard to train, we rely on a patch-based approach as generally used in the DL literature for denoising tasks \([8]\). The dictionary is learned over
Fig. 2: \( N \) patches \((\sqrt{d} \times \sqrt{d})\) are extracted from a eigen-image \( \hat{U}_k \) to form the training data matrix \( \mathbf{Y} \), which is used to learn a dictionary \( \mathbf{D} \) and a sparse reconstruction matrix \( \mathbf{\Gamma} \).

small patches of the noisy image itself, making the algorithm completely self-contained.

Given an eigen-image \( \hat{U}_k \in \mathbb{R}^{n_1 \times n_2} \) we (sub)sample \( N \) equally-spaced training patches with dimensions \((\sqrt{d} \times \sqrt{d})\). The patches are vectorized and stacked as the columns of a training matrix \( \mathbf{Y} \in \mathbb{R}^{d \times N} \). The corresponding dictionary \( \mathbf{D} \in \mathbb{R}^{d \times m} \) with \( m \geq d \) and sparse coefficient matrix \( \mathbf{\Gamma} \in \mathbb{R}^{m \times N} \) are obtained by jointly optimizing the following cost function:

\[
\{\hat{\mathbf{D}}, \hat{\mathbf{\Gamma}}\} = \arg \min_{\mathbf{D}, \mathbf{\Gamma}} \| \mathbf{\Gamma} - \mathbf{\Gamma} \|^2_F + g(\mathbf{\Gamma}) \tag{4}
\]

where \( g(\cdot) \) is a sparsity-inducing penalty function (for instance an \( \ell_0 \) or \( \ell_1 \) norm regularization on the columns of \( \mathbf{\Gamma} \)). Figure 2 summarizes the data structure and adopted notation.

The proposed denoising approach is described in Algorithm 1. Note that a new dictionary is learned for each eigen-image \( \hat{U}_k \). To fully characterize the algorithm, some elements are still to be defined: the rank and noise estimation, sparse coding, dictionary update and denoising functions (respectively in lines 4, 5, 13, 14 and 16) are further detailed in sections 3.3 and 3.4.

Algorithm 1: Dictionary-based HSI denoising approach

1. **INPUT:** Noisy hyperspectral image \( \mathbf{H} \)
2. \( \triangleright \) Low-rank step
3. \[ (\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) = \text{SVD}(\mathbf{H}) \]
4. \( r = \text{EstimateRank}(\mathbf{\Sigma}) \)
5. \( [\sigma_1, \ldots, \sigma_r] = \text{EstimateNoise}(\mathbf{\Sigma}) \)
6. \( \hat{\mathbf{U}} = \mathbf{U}_{(:, 1 : r)}, \quad \hat{\mathbf{V}} = \mathbf{V}_{(:, 1 : r)} \mathbf{\Sigma}(1 : r, 1 : r) \)
7. \( \triangleright \) Sparse step
8. for \( k = [1, \ldots, r] \) do
9. Extract \( N \) patches from \( \hat{\mathbf{U}}_k \) to form \( \mathbf{Y} \)
10. \( \triangleright \) Dictionary learning
11. Initialize \( \mathbf{D}_0 \)
12. for \( i = [1, \ldots, n_d] \) do
13. \( \mathbf{\Gamma}_i = \text{SparseCoding}(\mathbf{Y}, \mathbf{D}_{i-1}, \sigma_k) \)
14. \( \mathbf{D}_i = \text{DictionaryUpdate}(\mathbf{Y}, \mathbf{\Gamma}_i) \)
15. end for
16. \( \hat{\mathbf{U}}_k = \text{Denoise}(\hat{\mathbf{U}}_k, \mathbf{D}_{n_z}, \sigma_k) \)
17. end for
18. \( \hat{\mathbf{U}} = [\hat{\mathbf{U}}_1, \ldots, \hat{\mathbf{U}}_r] \)
19. **OUTPUT:** Recovered image \( \hat{\mathbf{X}} = \hat{\mathbf{U}} \hat{\mathbf{V}}^T \)

3.3. Dictionary Learning Algorithms

The customary DL framework to tackle problem (4) consists in iteratively alternating two optimization steps:

1. Sparse coding: updating \( \mathbf{\Gamma} \) with fixed \( \mathbf{D} \).
2. Dictionary update: optimizing for \( \mathbf{D} \) with fixed \( \mathbf{\Gamma} \).

**Sparse coding** There is an extensive literature on how to address this problem [13]. Although in theory any of the existing algorithms would serve our purposes, we use the traditional Orthogonal Matching Pursuit (OMP) algorithm [15] which allows us to directly control the reconstruction precision of the input data. As a stopping criterion, we will use a threshold \( \tau \) on the reconstruction error, i.e. the energy of the residual, proportional to the noise standard deviation \( \sigma \) (see Section 3.4 for more details on the noise level estimation) \( \tau = a \sigma \sqrt{d} \), with \( a = 1.16 \) an empirically-set gain following [8].

**Dictionary update** This is the step which distinguishes one DL algorithm from another. Two dictionary learning approaches are used:

1. K-SVD [9]: the columns of \( \mathbf{D} \) are updated one at a time via a rank-one SVD approximation of the error matrix with respect to the current atom.
2. SuKro [10]: it learns structured dictionaries which can be written as a sum of a few Kronecker products (denoted \( \boxtimes \)) of smaller matrices \( \mathbf{D}_{j,q} \in \mathbb{R}^{\sqrt{d} \times \sqrt{m}} \).

For the SuKro dictionary update step we use an Alternating Least-Squares (ALS) procedure where the blocks \( \mathbf{D}_{j,q} \) are optimized alternatively with a closed-form optimal update in each step. This algorithm, which is more efficient than the one originally proposed in [10], was proposed in [16] for a higher-order generalization of SuKro, i.e. for sums of Kronecker products of any number of blocks, and we apply it for the particular case of two blocks described in (5).

\[
\mathbf{D} = \sum_{q=1}^{n_{kro}} \mathbf{D}_{1,q} \boxtimes \mathbf{D}_{2,q}. \tag{5}
\]

**Computation of the denoised image** Once the learning process is finished, the denoised version of the eigen-image \( \hat{\mathbf{U}}_k \) is computed as follows. All possible patches from the noisy image –i.e. not only the \( N \) training patches, but all \((n_1 - \sqrt{d} + 1)(n_2 - \sqrt{d} + 1)\) overlapping patches with a one-pixel translation from its neighbors– are reconstructed with the learned dictionary using OMP with the same threshold \( \tau \). Each pixel in the final denoised image is then calculated as the weighted average of all reconstructed patches covering that pixel (each with weight one) and its value in the noisy image (with weight \( \lambda = (\sigma \sqrt{m})^{-1} \)). See [8] for details on why, in some sense, this is an optimal recovering procedure.
where $\epsilon$ is a threshold empirically calibrated to $\epsilon = 3 \times 10^{-2}$.

4. EXPERIMENTAL RESULTS

Four hyperspectral images were used in the experiments with $(n_1 \times n_2 \times p)$ pixels: San Diego (400 $\times$ 400 $\times$ 158), Houston (349 $\times$ 1905 $\times$ 144), Washington DC (1280 $\times$ 256 $\times$ 191), Urban (307 $\times$ 307 $\times$ 162). The images were corrupted with additive white Gaussian noise with standard deviation $\sigma$ uniform over all spectral modes and adjusted to give the desired input signal-to-noise ratio (SNR).

For each eigen-image (columns of $U_k$), we collect $N = 20000$ patches with $(6 \times 6)$ pixels from the noisy image to form the training data. As a pre-processing, each patch is centralized to zero mean in its pixel values. The patches are taken uniformly-spaced and partially overlapping. A dictionary is learned from this data with $n_{kron} = 20$ iterations of alternating optimization. Dictionaries are initialized with an Overcomplete (cropped) 2D-DCT (ODCT). In the case of SuKro, the remaining terms are initialized with unit-norm Gaussian random matrices, with no relevant impact on its performance.

The results reported in Table 1 are averaged over 10 runs with independent noise realizations. Standard deviations are usually around 0.005dB (and never bigger than 0.02dB).

Three variations of the proposed method are tested, with three different types of dictionary: K-SVD unstructured learned dictionary, SuKro learned structured dictionary with $n_{kron} = 3$ terms and an overcomplete but fixed ODCT dictionary (i.e. the learning process is skipped). The proposed methods are compared to: a Wavelet 3D approach [18] which uses only the sparsity prior by applying a 3D-wavelet basis directly in the data cube $H$; and the HyRes [3] method, which combines both sparse and low-rank assumptions. The latter is a state-of-the-art approach (see [3] for a detailed comparison of this method with other methods in the literature).

The proposed methods with K-SVD and SuKro dictionaries consistently outperform the state-of-the-art HyRes approach by about 1.5dB. A performance gain of about 1.3dB is already obtained by replacing the orthogonal Wavelet basis in HyRes by a fixed overcomplete DCT dictionary allied to the patch-based procedure. The learning process then brings an additional 0.2dB in denoising performance. A visual comparison is provided in Figure 4.

Execution times are reported in Table 2. The literature methods HyRes and Wavelet 3D take about one and two minutes respectively, independently of the noise level. The running times for the proposed method, on the other hand, increase as the noise level decreases. That's because the OMP reconstruction threshold gets smaller, requiring more iterations. The execution times range from 120 to 760s in the ODCT case and from 260 to 1600s with a learned dictionary (actually, SuKro is slightly faster than K-SVD). Therefore, the proposed technique with dictionary learning takes around 5 to 30 times longer than HyRes.
Table 1: Output SNR [dB] comparison with literature

<table>
<thead>
<tr>
<th>Method</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wav. 3D</td>
<td>20.19</td>
<td>23.36</td>
<td>26.51</td>
<td>29.48</td>
<td>32.33</td>
<td>35.08</td>
</tr>
<tr>
<td>HyRes</td>
<td>23.26</td>
<td>26.37</td>
<td>29.63</td>
<td>32.61</td>
<td>35.44</td>
<td>37.92</td>
</tr>
<tr>
<td>ODCT</td>
<td>24.79</td>
<td>27.77</td>
<td>30.79</td>
<td>33.77</td>
<td>36.78</td>
<td>38.88</td>
</tr>
<tr>
<td>K-SVD</td>
<td>24.92</td>
<td>27.86</td>
<td>30.89</td>
<td>33.88</td>
<td>36.68</td>
<td>39.00</td>
</tr>
<tr>
<td>SuKro</td>
<td><strong>24.95</strong></td>
<td><strong>27.93</strong></td>
<td><strong>30.94</strong></td>
<td><strong>33.89</strong></td>
<td><strong>36.69</strong></td>
<td><strong>38.99</strong></td>
</tr>
</tbody>
</table>

(a) Original (b) Noisy: 10dB (c) SuKro (d) HyRes

Fig. 4: Zoom in Washington image (100th spectral band).

Table 2: Execution times (in seconds) for Washington image

<table>
<thead>
<tr>
<th>Method</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wav. 3D</td>
<td>125</td>
<td>120</td>
<td>125</td>
<td>120</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>HyRes</td>
<td>55</td>
<td>50</td>
<td>50</td>
<td>49</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>ODCT</td>
<td>128</td>
<td>201</td>
<td>296</td>
<td>408</td>
<td>533</td>
<td>761</td>
</tr>
<tr>
<td>K-SVD</td>
<td>273</td>
<td>445</td>
<td>677</td>
<td>980</td>
<td>1315</td>
<td>1650</td>
</tr>
<tr>
<td>SuKro</td>
<td>262</td>
<td>394</td>
<td>592</td>
<td>857</td>
<td>1143</td>
<td>1433</td>
</tr>
</tbody>
</table>

5. CONCLUSION

In this paper, we proposed to replace fixed sparsifying transforms by overcomplete learned dictionaries to improve state-of-the-art performance in the HSI denoising task.

Although the proposed patch-based approach is markedly slower than other previous methods, running times remain reasonable for an offline application, even for large images. A potential improvement would be to also denoise the right term of the low-rank decomposition (spectral component V).

6. REFERENCES


