

This item is the archived peer-reviewed author-version of:

Hyperspectral image compression optimized for spectral unmixing

Reference:

Karami Robati Azam, Heylen Rob, Scheunders Paul.- Hyperspectral image compression optimized for spectral unmixing
IEEE transactions on geoscience and remote sensing / Institute of Electrical and Electronics Engineers [New York, N.Y.] - ISSN
0196-2892 - (2016), p. 1-11

Full text (Publishers DOI): <http://dx.doi.org/doi:10.1109/TGRS.2016.2574757>

HYPERSPECTRAL IMAGE COMPRESSION OPTIMIZED FOR SPECTRAL UNMIXING

Azam Karami^{a,b}, Rob Heylen^a, Paul Scheunders^a

^aVisionlab, University of Antwerp, Belgium

^b Faculty of Physics,, Shahid Bahonar University of Kerman, Kerman, Iran

ABSTRACT

In this paper, we present a new lossy compression method for hyperspectral images that aims to optimally compress in both spatial and spectral domains and simultaneously minimizes the effect of the compression on linear spectral unmixing performance. To achieve this, a non-negative tucker decomposition is applied. This decomposition is a function of three dimension parameters. By employing a link between this decomposition and the linear spectral mixing model, an optimization problem is defined to find the optimal parameters by minimizing the root mean square error between the abundance matrices of the original and reconstructed datasets. The resulting optimization problem is solved by a Particle Swarm Optimization algorithm. An approximate method for fast estimation of the free parameters is introduced as well. Our simulation results show that, in comparison with well-known state-of-the-art lossy compression methods, an improved compression as well as spectral unmixing performance of the reconstructed hyperspectral image is obtained. It is noteworthy to mention that the superiority of our method becomes more apparent as the compression ratio grows.

Index Terms— Lossy Compression, Hyperspectral Images, Non-negative Tucker Decomposition, Spectral Unmixing

1. INTRODUCTION

Hyperspectral images (HSI) are usually volumetric and require a lot of space and time for archiving and transferring. For instance, most of the modern hyperspectral sensors such as AVIRIS¹, EnMAP² and HYPERION³ capture more than 200 spectral bands covering the spectrum in a range of wavelengths between $0.4\mu m$ and $2.5\mu m$. Future sensors will contain even more bands, e.g. the IASI⁴ sensor will capture 8461 spectral bands covering the range $3.62-15.5\mu m$. The number of pixels composing HSI is usually high. The high spatial and spectral dimensionality of the HSI makes their analysis computationally very costly. Therefore, to reduce the cost of

storage equipment and bandwidth, compression of HSI has received particular attention in recent years.

There are two types of redundancy in HSI: spatial and spectral redundancies. These redundancies allow for the development of effective compression algorithms. Traditional HSI compression algorithms such as dimensionality reduction through Principal Component Analysis (PCA) and Independent Component Analysis were mainly based on vectorizing the images, not considering the spatial correlation [1]. Recent methods consider HSI as a 3D dataset with two spatial and one spectral dimension.

Several 3D wavelet-based techniques such as the Set Partitioning in Hierarchical Trees (SPIHT) algorithm and the Set Partitioned Embedded bloCK (SPECK) algorithm have been proposed in [2]. It was demonstrated that 3D-SPECK outperforms 3D-SPIHT in compression efficiency. In another attempt, an optimal 3D anisotropic wavelet decomposition for HSI compression was introduced [3]. However, the results showed that a 3D isotropic wavelet is not suitable for HSI since HSI are spectrally more correlated than spatially. In [4], a hybrid 3D wavelet transform for spectral and spatial decorrelations combined with JPEG2000 has been proposed. The results showed an improved performance with respect to the SPIHT and SPECK algorithms. In [5], JPEG2000 was combined with PCA, in which a reduced number of Principal Components are retained and coded. The results reveal an improved performance over the use of the wavelet transform. Finally, other 3D decomposition methods can be applied, as in [6] where HSI were treated as a 3D tensor and a Non-negative Tucker Decomposition (NTD) was applied.

After compression and transmitting, a HSI is reconstructed to be used for a specific analysis task in a real application such as classification, spectral unmixing, target detection etc. In recent years, the effect of lossy compression on the mentioned applications has been investigated. In [7], a method based on an integration of PCA and the discrete cosine transform (DCT) was developed with the aim of trying to minimize the effect of the compression on target detection. Several HSI compression methods with the aim of anomaly preservation were proposed as well. In [8], anomalies are first detected and then sent in lossless mode while the remainder of the image is encoded using PCA+JPEG2000.

In [9], the effect of compression on classification, spectral

¹<http://aviris.jpl.nasa.gov>

²<http://www.enmap.org/>

³<http://eo1.usgs.gov/sensors/hyperion>

⁴<http://wdc.dlr.de/sensors/iasi/>

unmixing and anomaly detection was investigated, using a 3D lossy compression method based on a hybrid approach using PCA, wavelets and JPEG2000. While classification performances remained unaffected, the performance of spectral unmixing and anomaly detection were seriously degraded by the compression. A larger study using different compression methods in [10] revealed similar results. In [11], the effect on spectral unmixing was studied using the H.264/AVC video coding standard. For high compression ratio (CR) the unmixing performances were highly reduced. In [12], it was proposed to use operational bit rates to balance between high compression ratios and classification and spectral unmixing performance.

All these studies show that lossy compression has a serious impact on the spectral unmixing performance. The reason is that lossy compression and spectral unmixing serve opposite purposes. For compression, the high spectral redundancies are optimally reduced for maximal compression ratios. Unmixing requires the optimal preservation of subtle spectral differences to find the true endmembers and to obtain accurate abundances. In [13], compression is proposed by unmixing the HSI using the linear spectral mixing model, and then transmitting the endmember and abundance matrices. At the decoder, the HSI is then reconstructed by inverting the unmixing model. Although, by construction, optimal spectral unmixing performance is obtained after reconstruction, the achievable CR is rather moderate, and not competitive with state of the art compression methods.

In this work, we rather look for a method that optimizes both purposes simultaneously. First, we need a compact representation that is flexible in the choice of spatial and spectral dimensions. Although wavelets are proven to be superior for spatial compression, they are inferior to methods based on eigenvalue decompositions in the spectral dimension. From these methods, the NTD has the required flexibility [6]. Then, we will make the unmixing to become part of the compression process by establishing a direct link between the NTD representation and the linear spectral mixing model. This allows to derive an optimization problem that optimizes the dimension parameters for an optimal compression and an optimal preservation of the abundance maps simultaneously.

In the proposed method, the original dataset is treated as a 3D tensor, and spatially partitioned into smaller subtensors, after which the NTD is applied to each subtensor. The NTD contains three free parameters which are the rank values of the core tensor, and which determine the obtainable CR. We introduce a new optimization approach to select these free parameters, using a cost function that is the mean square error (MSE) between the abundance matrices of the original and reconstructed HSI, hereby efficiently optimizing jointly the Signal-to Noise Ratio (SNR) and the spectral unmixing performance of the reconstructed HSI. The optimization is performed using a Particle Swarm Optimization (PSO) method.

In the experimental section, the proposed method is com-

pared to the well-known state-of-the-art techniques; the 3D-SPECK algorithm from [2] and PCA+JPEG2000 from [5]. The performance is evaluated on synthetic and real datasets. The experimental results demonstrate that our algorithm provides a much smaller MSE and higher SNR variance for the desired CR, especially when the target CR is high. The remaining of the paper is organized as follows: section 2 introduces the proposed algorithm. Experiments are performed in section 3, and section 4 describes the concluding remarks.

2. PROPOSED ALGORITHM

2.1. Non-Negative Tucker Decomposition

Let us denote the HSI as a 3-rd order tensor $\underline{\mathbf{X}} \in \mathbf{R}^{I_1 \times I_2 \times I_3}$ where I_1, I_2 are the spatial dimensions and I_3 is the number of bands. The third-order Tucker decomposition is a decomposition of $\underline{\mathbf{X}}$ into an unknown non-negative core tensor $\underline{\mathbf{G}} \in \mathbf{R}^{J_1 \times J_2 \times J_3}$ multiplied by a set of three unknown non-negative matrices $\mathbf{B}^{(n)} = [b_1^{(n)}, b_2^{(n)}, \dots, b_{J_n}^{(n)}] \in \mathbf{R}^{I_n \times J_n}$ ($n = 1, \dots, 3$) [14, 15] (see Fig. 1):

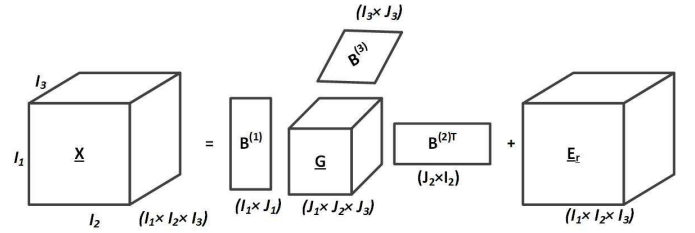


Fig. 1: Third-order Tucker decomposition

$$\begin{aligned}
 \underline{\mathbf{X}} &= \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \sum_{j_3=1}^{J_3} g_{j_1 j_2 j_3} b_{j_1}^{(1)} \circ b_{j_2}^{(2)} \circ b_{j_3}^{(3)} + \underline{\mathbf{E}}_r \\
 &= \underline{\mathbf{G}} \times_1 \mathbf{B}^{(1)} \times_2 \mathbf{B}^{(2)} \times_3 \mathbf{B}^{(3)} + \underline{\mathbf{E}}_r \\
 &= \underline{\mathbf{G}} \times \{\mathbf{B}\} + \underline{\mathbf{E}}_r \\
 &= \hat{\underline{\mathbf{X}}} + \underline{\mathbf{E}}_r
 \end{aligned} \tag{1}$$

where tensor $\hat{\underline{\mathbf{X}}}$ is an estimation of tensor $\underline{\mathbf{X}}$, $\bar{J} = [J_1, J_2, J_3]$ are the dimensions of the core tensor $\underline{\mathbf{G}}$ and $\underline{\mathbf{E}}_r$ denotes the estimation error. The objective is to find the optimal component matrices $\mathbf{B}^{(n)} \in \mathbf{R}^{I_n \times J_n}$ and the core tensor $\underline{\mathbf{G}} \in \mathbf{R}^{J_1 \times J_2 \times J_3}$ for a given \bar{J} :

$$\begin{aligned}
 (\mathbf{B}^{(n)*}, \underline{\mathbf{G}}^*) &= \arg \min_{\mathbf{B}^{(n)} \geq 0, \underline{\mathbf{G}} \geq 0} \left(\frac{1}{2} \left\| \underline{\mathbf{X}} - \hat{\underline{\mathbf{X}}} \right\|_F^2 \right) \tag{2} \\
 \hat{\underline{\mathbf{X}}}^*(\bar{J}) &= \underline{\mathbf{G}}^* \times \{\mathbf{B}^*\}
 \end{aligned}$$

The CR is the ratio between the total number of bits $N_{\underline{\mathbf{X}}}$ of the original input image and the number of bits of the com-

pressed dataset:

$$CR(\bar{J}) = \frac{N_{\underline{\mathbf{X}}}}{\sum_{n=1}^3 N_{\mathbf{B}^{(n)}} + N_{\underline{\mathbf{G}}}} \quad (3)$$

with $N_{\underline{\mathbf{X}}} = I_1 \times I_2 \times I_3 \times \lceil \log_2(\max \underline{\mathbf{X}}) \rceil$, $N_{\mathbf{B}^{(n)}} = I_n \times J_n \times \lceil \log_2(\max \mathbf{B}^{(n)}) \rceil$ and $N_{\underline{\mathbf{G}}} = J_1 \times J_2 \times J_3 \times \lceil \log_2(\max \underline{\mathbf{G}}) \rceil$ after rounding all pixel values to integers, $\lceil \cdot \rceil$ denotes rounding to the next integer. After obtaining the optimal matrices $\mathbf{B}^{*(1)}, \mathbf{B}^{*(2)}, \mathbf{B}^{*(3)}$ and the core tensor $\underline{\mathbf{G}}^*$ they are coded. Because the matrices and in particular the core tensor are very sparse, they contain a lot of zero elements. In the proposed method, all elements of the matrices and core tensor are first rounded to two decimals after which arithmetic coding is applied. The $CR(\bar{J})$ after coding is then obtained as follow:

$$CR(\bar{J}) = \frac{N_{\underline{\mathbf{X}}}}{\sum_{n=1}^3 N'_{\mathbf{B}^{(n)}} + N'_{\underline{\mathbf{G}}}} \quad (4)$$

where $N'_{\underline{\mathbf{G}}}$ and $N'_{\mathbf{B}^{(n)}}$ are the total number of bits, required to transmit each matrix and core tensor respectively after applying arithmetic coding. The CR is directly dependent on the values $\bar{J} = [J_1, J_2, J_3]$.

Rank estimation of the Tucker decomposition of hyper-spectral images has been done before [16, 17]. Optimal values of the rank of the tensor were based on optimizing eigenvalue decompositions of the image correlation or covariance matrices and the obtained \bar{J} corresponded to one particular CR. However in practical applications, specific bandwidth limitations may require a specific CR. In this work, we want to obtain the optimal rank for any given CR. Therefore, the following optimization problem is considered:

$$\begin{aligned} \min. \quad & MSE(\bar{J}) \\ \text{st} : \quad & \begin{cases} 1 < J_1 < I_1 \\ 1 < J_2 < I_2 \\ 1 < J_3 < I_3 \\ CR(\bar{J}) = Const \end{cases} \end{aligned} \quad (5)$$

with:

$$MSE(\bar{J}) = \left(\frac{1}{I_1 I_2 I_3} \right) \sum_{k=1}^{I_3} \sum_{j=1}^{I_2} \sum_{i=1}^{I_1} (x_{ijk} - \hat{x}_{ijk}^*)^2 \quad (6)$$

Here, the optimal values for $\bar{J} = [J_1, J_2, J_3]$ minimize the MSE value for any given compression ratio.

2.2. Link between the NTD and spectral unmixing

In this section, a relation between the NTD and spectral unmixing is established. Let us unfold the original dataset $\underline{\mathbf{X}} \in \mathbf{R}^{I_1 \times I_2 \times I_3}$ in the third direction as $\mathbf{X}_{(3)} \in \mathbf{R}^{I_3 \times I}$ with the number of pixels $I = I_1 I_2$. When applying the linear spectral unmixing model (LSU):

$$\begin{aligned} \mathbf{X}_{(3)} &= \mathbf{E}\mathbf{A} + \mathbf{N} \\ \mathbf{N} &= \mathcal{N}(0, \sigma^2) \end{aligned} \quad (7)$$

where $\mathbf{E} \in \mathbf{R}^{I_3 \times q}$ is the endmember matrix and $\mathbf{A} \in \mathbf{R}^{q \times I}$ is the abundance matrix, q is the number of endmembers. $\mathbf{N} \in \mathbf{R}^{I_3 \times I}$ is zero mean Gaussian noise. Usually in LSU some constraints are considered. The endmember and abundance matrices should be non-negative and each column of the abundance matrix sums up to one.

As described in the introduction, different studies [10–12] have shown that the compression of a HSI has a large impact on the spectral unmixing performance. In [10], it was shown that, due to the spectral distortions occurring during compression, the effect of compression is more destructive on the endmembers than on the abundance maps. Therefore, in this work, we will concentrate on the development of a compression method that optimally preserves the abundance maps. Sometimes, some information about the endmembers is known in advance, e.g. as spectra collected on the ground by a field spectroradiometer or extracted from a library, such as the USGS library. If there is no information, the number of endmembers needs to be estimated first, e.g. using the HySime algorithm [18], after which an endmember extraction algorithm, e.g. vertex component analysis (VCA) [19] is applied. The endmember locations are extracted and along with the endmember spectra transmitted. The elements of \mathbf{E} are integer numbers (usually 16-bit) as are their locations. The arithmetic coding is applied in lossless mode, since it is not expected that high compression ratio's are obtained in lossy mode due to the fact that the endmembers are generally uncorrelated. After sending the compressed HSI, the endmembers and their locations to the receiver, the HSI is decompressed and the original endmembers are induced at their locations, after which the abundance matrix $\hat{\mathbf{A}}$ is calculated. Here we use a Fully Constrained Least Squares Unmixing (FCLSU) method [20].

The reconstructed dataset $\hat{\mathbf{X}}$ can also be unfolded in the third direction $\hat{\mathbf{X}}_{(3)}$, and modeled using the transmitted endmember matrix \mathbf{E} as:

$$\hat{\mathbf{X}}_{(3)} \approx \mathbf{E}\hat{\mathbf{A}} \quad (8)$$

Metricized in the third direction, Eq. 2 contains the term:

$$\begin{aligned} \min \left(\frac{1}{2} \left\| \mathbf{X}_3 - \hat{\mathbf{X}}_3 \right\|_F^2 \right) &\approx \min \left(\frac{1}{2} \left\| \mathbf{E}\mathbf{A} - \mathbf{E}\hat{\mathbf{A}} \right\|_F^2 \right) \\ &\sim \min \left(\frac{1}{2} \left\| \mathbf{A} - \hat{\mathbf{A}} \right\|_F^2 \right) \end{aligned} \quad (9)$$

Because the endmember matrix is constant, the MSE between the abundance matrices of the original and the reconstructed datasets is minimized. The MSE in Eq. 5 is now defined as:

$$MSE(\bar{J}) = \left(\frac{1}{I_1 I_2 q} \right) \sum_{k=1}^q \sum_{j=1}^{I_2} (a_{jk} - \hat{a}_{jk}^*)^2 \quad (10)$$

In order to limit the search space in Eq.(5), some constraints can be considered, based on the correlation properties

of HSI. The spectral correlation is higher than the spatial correlation. Therefore, the choice of J_3 has a significant effect on the CR. Since we have assumed that the HSI is modeled by the linear spectral mixing model, the value for J_3 does not need to be higher than the number of endmembers q . Since the spatial correlation is low, the values of J_1 and J_2 should not be chosen too low. The optimization problem will be constrained to:

$$\begin{aligned} \min . \quad & MSE(\bar{J}) \\ \text{st} : \quad & \begin{cases} I_1/2 < J_1 < I_1 \\ I_2/2 < J_2 < I_2 \\ 1 < J_3 < q \\ CR(\bar{J}) = Const \end{cases} \end{aligned} \quad (11)$$

This is a nonlinear optimization problem and can be converted to:

$$\begin{aligned} \min . \quad & MSE(\bar{J}) + \lambda \|CR(\bar{J}) - const\| \\ \text{st} : \quad & \begin{cases} I_1/2 < J_1 < I_1 \\ I_2/2 < J_2 < I_2 \\ 1 < J_3 < q \end{cases} \end{aligned} \quad (12)$$

where λ can vary between zero and one. If λ is selected close to zero, the algorithm tries to minimize $MSE(\bar{J})$. This however does not guarantee that the obtained CR will be close to the required value. Therefore, in the proposed method, λ is selected near one.

2.3. Particle Swarm Optimization Technique

Since J_1, J_2 and J_3 are the rank values of the core tensor, they are integer-valued. This makes Eq.(12) an integer programming problem, for which a heuristic method is appropriate. In this work, we choose to use Particle Swarm Optimization (PSO), of which the time complexity is very competitive to that of other heuristic methods [21]. PSO tries to minimize the objective function ($MSE(\bar{J}) + \lambda \|CR(\bar{J}) - const\|$) by finding the optimum values for \bar{J} in the search space ($I_1/2 < J_1 < I_1, I_2/2 < J_2 < I_2$ and $1 < J_3 < q$). The PSO algorithm will be briefly explained in the following.

PSO is a stochastic population-based optimization approach, first proposed in [22]. The algorithm defines a population of possible solutions as positions of particles in the search space. At each iteration, the particles move around and their positions and velocities are updated and validated against the fitness function from Eq.(13). The updating is done according to:

$$\begin{aligned} \bar{J}^l(k+1) &= \lfloor \bar{J}^l(k) + \bar{V}^l(k+1) \rfloor \quad (13) \\ \bar{V}^l(k+1) &= \phi(k)\bar{V}^l(k) + \alpha_1 [\gamma_{1l} (\bar{P}_l - \bar{J}^l(k))] + \\ &\quad \alpha_2 [\gamma_{2l} (\bar{P}_g - \bar{J}^l(k))] \end{aligned}$$

where $\bar{J}^l(k)$ denotes the position of particle l at iteration k and is a possible solution, in our case a possible set of dimensions of the core tensor. $\bar{V}^l(k)$ is its corresponding velocity.

The rounding operator $\lfloor \cdot \rfloor$ ensures that the vector $\bar{J}^l(k+1)$ is integer-valued. γ_{1l}, γ_{2l} are random numbers in the interval $[0,1]$. \bar{P}_l is particle l 's best position so far and \bar{P}_g is the best position of the whole swarm so far. $\phi(k)$ is a decreasing linear inertia function [23]:

$$\phi(k) = \phi_{\max} - (\phi_{\max} - \phi_{\min}) \times \frac{k}{N_{th}} \quad (14)$$

where N_{th} is the maximum number of iteration and ϕ_{\max} and ϕ_{\min} are usually set to 0.9 and 0.4 respectively.

α_1 and α_2 are two positive constants called acceleration coefficients; α_1 controls the movement of the particle, searching around for its best location and α_2 controls the influence of the swarm on the particle's behavior. For PSO convergence, the following condition should be considered [22]:

$$0 \leq \alpha_1 + \alpha_2 \leq 4 \quad (15)$$

A detailed description of the position and velocity update scheme can be found in [22], which also contains a detailed analysis of the impact of the different setting parameters and a derivation of a set of necessary and sufficient conditions that ensure stable behavior of the algorithm and guarantee convergence. The algorithm works as follows:

1. Initial population generation: Initial positions $\bar{J}^l(0)$ and velocities $\bar{V}^l(0)$ of the particles are randomly generated throughout the search space bounded by specific limits on each variable as given by Eq.(12) [24]:

$$\begin{aligned} \bar{J}^l(0) &= \lfloor \bar{J}_{\min} + r_1(\bar{J}_{\max} - \bar{J}_{\min}) \rfloor \quad (16) \\ \bar{V}^l(0) &= \bar{J}_{\min} + r_2(\bar{J}_{\max} - \bar{J}_{\min}) \end{aligned}$$

where r_1, r_2 are random vectors with components uniformly distributed in $[0, 1]$. The population consists of N_p particles.

2. Fitness function evaluation: For each particle, the objective function value $MSE(\bar{J}^l(k)) + \lambda \|CR(\bar{J}^l(k)) - const\|$ is evaluated.
3. Updating: First, the optimum particle positions \bar{P}_l and the global optimum particle position \bar{P}_g are updated. Then, the positions and velocities of each particle are updated using Eq.13.
4. Stopping criterion: Repeat steps 2 & 3 until the change in the fitness function is smaller than a specified tolerance:

$$|MSE(\bar{J}^l(k+1)) + \lambda \|CR(\bar{J}^l(k+1)) - const\| - MSE(\bar{J}^l(k))| \quad (17)$$

Here, ϵ is chosen to be 10^{-5} . Finally, the best global position, \bar{P}_g is used as the best estimate of the tensor rank in our current application:

$$\bar{J}^* = \bar{P}_g = [P_{1g}, P_{2g}, P_{3g}] \quad (18)$$

After obtaining the optimal values of \bar{J}^* , the optimal matrices $\mathbf{B}^{*(1)}, \mathbf{B}^{*(2)}, \mathbf{B}^{*(3)}$ and core tensor \mathbf{G}^* are calculated using NTD, which solves Eq.(2). Then, arithmetic coding is applied to them. Pseudo-code for the PSO-NTD algorithm is given in Algorithm 1:

Algorithm 1 PSO-NTD Algorithm

- 1- Initialize the PSO parameters
 - 2- Generate N_p particles with random positions and velocities
 - 3- Calculate $\mathbf{B}^{*(1)}, \mathbf{B}^{*(2)}, \mathbf{B}^{*(3)}, \mathbf{G}^*$ using NTD
 - 4- Encode $\mathbf{B}^{*(1)}, \mathbf{B}^{*(2)}, \mathbf{B}^{*(3)}, \mathbf{G}^*$ using arithmetic coding
 - 5- Calculate the fitness function of Eq. (12)
 - 6- Stopping criterion satisfied?
Yes : Go to step 7
No: Update the positions and velocities of the particles using Eq.(13) and go back to step 3
 - 7- Return the optimal values of (J_1, J_2, J_3)
 - 8- Calculate $\mathbf{B}^{*(1)}, \mathbf{B}^{*(2)}, \mathbf{B}^{*(3)}, \mathbf{G}^*$ using NTD
 - 9- Encode $\mathbf{B}^{*(1)}, \mathbf{B}^{*(2)}, \mathbf{B}^{*(3)}, \mathbf{G}^*$, the endmembers and their locations using arithmetic coding
 - 10- Transmit the compressed dataset
 - 11- Reconstruct the dataset
-

2.4. PSO-subNTD

The complexity of the PSO-NTD algorithm is a function of the three dimension parameters (I_1, I_2, I_3) of the HSI. Since the spatial correlation of HSI is much smaller than the spectral correlation, the original dataset can be partitioned spatially into smaller subtensors with size (I'_1, I'_2, I_3) (See Fig.2). When the PSO-NTD is applied to each subtensor separately, the computational load is decreased. We will refer to this algorithm as PSO-subNTD.

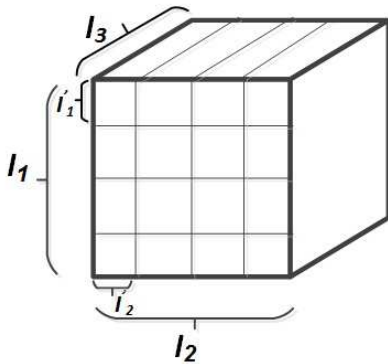


Fig. 2: Spatial partitioning in subtensors

2.5. Fast subNTD

The most time consuming step is the PSO optimization which scales linearly with the number of particles and the number of iterations. This method is too complex, when applied for online compression. The optimization step can be avoided when fixed values for the core tensor rank would be available. In this case, the obtained \bar{J} is not optimal and a fixed CR is achieved.

Therefore, we propose an alternative fast subNTD method by selecting fixed values for the core tensor rank. Because the spatial correlation is low, the spatial dimensions are selected as:

$$\begin{aligned} J'_1 &= \left\lfloor 0.95 \times I'_1 \right\rfloor \\ J'_2 &= \left\lfloor 0.95 \times I'_2 \right\rfloor \end{aligned} \quad (19)$$

J_3 is selected by applying the HySime algorithm to each subtensor, and J_3 is selected as being equal the obtained number of endmembers q . NTD is then applied on each subtensor using this selection and arithmetic coding is applied to the obtained core tensors and matrices.

3. EXPERIMENTAL RESULTS

First we want to emphasize that the proposed method is used for compression and the aim is to minimize the effect of compression on spectral unmixing. In unmixing analysis, reflectance datasets are generally used because the available information about the endmembers is in reflectance format such as is the case in the USGS library [25]. To validate the compression performance of the proposed method, first two AVIRIS radiance datasets (Cuprite (first scene) and Moffett field) [26] are analyzed. Then, in order to investigate the impact of compression on spectral unmixing, the method is applied on reflectance data: one synthetic and two real well-known hyperspectral datasets "Cuprite" and "Hydix" [26, 27].

The compression is quantified in bit per pixel per band (bpppb). The compression ratio is then the ratio of the original dataset (usually 16 bpppb) to that of the obtained bpppb. The compression results are validated using the SNR variance between the original and reconstructed images:

$$\begin{aligned} SNR_{dB} &= 10 \times \log_{10} \left(\frac{var(\mathbf{X})}{MSE^*} \right) \\ MSE^* &= \sum_{k=1}^{I_3} \sum_{j=1}^{I_2} \sum_{i=1}^{I_1} (x_{ijk} - \hat{x}_{ijk})^2 \end{aligned} \quad (20)$$

In order to evaluate the unmixing performance, the MSE value between the original and reconstructed abundance maps using Eq.(10) is given.

Our method was developed using the tensor toolbox [28] and a PSO toolbox [29]. The HySime algorithm is applied to obtain the number of endmembers [18], VCA is used to extract the endmembers from the original dataset [19], and a fast version of FCLSU to calculate the abundance fraction matrix [20].

The population size of PSO was chosen to be 20. α_1 and α_2 are usually taken in the range between 1.5 and 2.5, satisfying the condition of Eq. 15. Simulation results showed that a fixed acceleration coefficient of 2 generated the best solutions. This value was also used in previous research (for example see [22] [23]). Since PSO is a stochastic process, the performance may vary by the specific choice of the initial conditions. Therefore, the optimization process is repeated twenty times and the average results are shown.

The proposed method is compared to two state of the art algorithms: the 3D-SPECK algorithm [2] and PCA+JPEG2000 [5]. In both algorithms, a wavelet transform is applied. For this, we use the popular biorthogonal 9/7 wavelet. Since the spectral correlation is higher than the spatial correlation, the wavelet decomposition level for the spectral dimension should be higher than for the spatial domain. After experimenting on the real images, we chose to apply 6 spatial and 8 spectral levels for the wavelet decomposition in the 3D-SPECK algorithm. 3D-SPECK is implemented using the QccPack toolbox [30]. The JPEG2000 coding is done using the Kakadu software with a quantization step size of 10^{-7} [31] [32]. The number of principal components (PCs) for PCA+JPEG2000 is also shown because its value affected the obtained SNR variance value.

3.1. Real Radiance Datasets

Two AVIRIS radiance datasets (Moffett field and Cuprite (first scene)) are used. These 16-bit radiance datasets have been cropped spatially to a size of 512×512 , and they are composed of 224 spectral bands. Before applying the proposed method the original datasets are partitioned into 64 subtensors ($I'_1 = 64, I'_2 = 64, I_3 = 224$). Since spectral unmixing is not part of the process, Eq. 5 is optimized using PSO. The obtained optimal values of \tilde{J}^* , the matrices $\mathbf{B}^{*(1)}, \mathbf{B}^{*(2)}, \mathbf{B}^{*(3)}$ and the core tensor $\underline{\mathbf{G}}^*$ are encoded using arithmetic coding. The compression results are shown in Tables 1 & 2. Compared with the other techniques, PSO-subNTD obtained the highest SNR variance values at different bpppb.

Table 1: SNR variance (dB) for the different compression methods on the Moffett radiance dataset

bpppb Method	0.05	0.1	0.25	0.5	1
3D-SPECK	25.32	29.23	35.09	39.82	45.16
PCA+JPEG2000	34.71 (PCs=14)	39.26 (PCs=25)	44.02 (PCs=39)	47.16 (PCs=82)	50.99 (PCs=104)
PSO-subNTD	37.06	42.64	46.31	49.89	52.77

Table 2: SNR variance (dB) for the different compression methods on the Cuprite radiance dataset

bpppb Method	0.05	0.1	0.25	0.5	1
3D-SPECK	35.14	38.76	43.09	46.5	50.72
PCA+JPEG2000	43.13 (PCs=20)	45.25 (PCs=23)	48.17 (PCs=38)	50.51 (PCs=75)	54.21 (PCs=100)
PSO-subNTD	46.17	48.96	51.96	53.74	56.83

It is noteworthy to mention that the radiance Cuprite dataset has well known calibration artifacts [33] and care should be taken to use it for compression tests as a specific coder may be designed that is optimized for these artifacts. Therefore, in the following a synthetic and two real reflectance Cuprite and Hydice datasets (without these artifacts) are also considered in order to show that the obtained coding performances are still consistent.

3.2. Synthetic Reflectance Dataset

To generate the endmember matrix \mathbf{E} , we randomly select a number of spectral signatures from the USGS mineral database [25]. The dimensionality is chosen to be 224, the number of endmembers is chosen to be 15. The abundance fraction matrix (\mathbf{A}) is generated by sampling from a Dirichlet distribution [18], including the non-negativity and the sum to one constraints. In this way, images of 128×128 pixels are constructed. Then, i.i.d. Gaussian noise is added, with different values of the variance. To apply the subtensor method, the image is partitioned into four subtensors ($I'_1 = 64, I'_2 = 64, I_3 = 224$). Then, HySime and VCA are applied to each subtensor separately in order to find the number of endmembers (q) and to identify the endmembers and their locations. Finally, PSO-subNTD is applied to the subtensors.

In Tables 3 & 4, the results of an experiment is shown using $\sigma^2 = 0.1$ and different bpppb varying from 0.05 to 1.

The proposed algorithm outperforms the two state of the art methods with respect to SNR variance and MSE.

Table 3: SNR variance(dB) values for different compression methods on the synthetic dataset with $\sigma^2 = 0.1$

bpppb Method	0.05	0.1	0.2	0.5	1
3D-SPECK	24.15	26.04	28.04	30.9	35.05
PCA+ JPEG2000	27.31 (PCs=23)	28.95 (PCs=27)	31.51 (PCs=43)	33.13 (PCs=84)	37.43 (PCs=107)
PSO-subNTD	32	34.04	38.1	39.13	41.21

Table 4: MSE for different compression methods on the synthetic dataset with $\sigma^2 = 0.1$

bpppb Method	0.05	0.1	0.2	0.5	1
3D-SPECK	0.2209	0.1849	0.0729	0.0441	0.0156
PCA+ JPEG2000	0.1444	0.0625	0.0441	0.0324	0.0149
PSO-NTD	0.0380	0.0289	0.0225	0.0196	0.0137

We also applied the compression method from [13], where a HSI is unmixed, and the obtained endmembers and abundances are transmitted to be reconstructed. For our synthetic dataset at bpppb=1, the obtained MSE is zero by construction and the obtained SNR variance was 27.3 dB, which is much smaller than obtained with the other compression methods.

3.3. Real Reflectance Datasets

The Cuprite Reflectance dataset is very popular for spectral unmixing. It is freely available in [26]. The file f970619t01p02.r02.sc04.a.rfi is used. This reflectance dataset is composed of $614 \times 512 \times 224$ pixels. We selected a subimage of 301×365 pixels (X coordinates 50:350, Y coordinates 250:614). This part was chosen since it contains most minerals of interest. The atmospheric and water absorption bands (bands [1-4],[107-113],[153-168],[217-224]) are removed from the original dataset prior to spectral unmixing and compression [20] [34]. The remaining 189 bands are used in the experiment. The image is partitioned into 30 subtensor (20 subtensors with size $64 \times 64 \times 189$, 5 subtensors with size $64 \times 45 \times 189$, 4 subtensors with size $45 \times 64 \times 189$ and one with size $45 \times 45 \times 189$) for (I'_1, I'_2, I'_3) , while the number of endmembers (q) and their locations are obtained using HySime and VCA for each subtensor separately.

Table 5 shows the obtained SNR variance values at different bpppb ratios for the three methods. PSO-subNTD outperforms the other methods, especially when the bpppb becomes lower than 0.5. Fig. 3 displays reconstructed band 170 at 0.2 bpppb. By visual comparison, the reconstructed image obtained from the proposed algorithm is less noisy and shows more details than the other reconstructed images.

In order to show what the separate contributions of the NTD and the arithmetic coding are on the different matrices and core tensor, we analyse more closely the result at a bpppb=0.05, which comes down to a CR of 320. If no arithmetic coding is applied, the obtained compression ratio is 61.45. The extra compression ratios due to the arithmetic coding are $CR_{B^{*(1)}} = 1.64$, $CR_{B^{*(2)}} = 1.6$, $CR_{B^{*(3)}} = 1.73$ and $CR_{G^*} = 12.7$ for the three matrices and the core tensor respectively. This demonstrates that the obtained core tensor is very sparse which is beneficial for the coding. The CR of the endmember matrix \mathbf{E} and the endmember locations is 1.31 and 1.27 respectively.

Table 5: SNR variance (dB) for the different compression methods on the reflectance Cuprite dataset

bpppb Method	0.05	0.1	0.25	0.5	1
3D-SPECK	25.74	28.77	32.22	35.02	38.69
PCA+ JPEG2000	28.31 (PCs=15)	31.02 (PCs=20)	34.08 (PCs=30)	36.79 (PCs=55)	40.28 (PCs=100)
PSO-subNTD	32.06	34.64	37.96	41.31	43.71

Table 6 shows the MSE between the abundance fraction matrix of the uncompressed and reconstructed datasets. The MSE values of the proposed method are smaller than of the other methods. Fig.4 plots 5 of the most significant endmembers of the Cuprite dataset. The obtained abundance maps at 0.2 bpppb are shown in Fig.5. The results demonstrate that the abundance maps of the proposed method are very similar to the original abundance maps. One can conclude that the proposed method simultaneously displays a good compression performance and minimizes the effect of lossy compression on spectral unmixing.

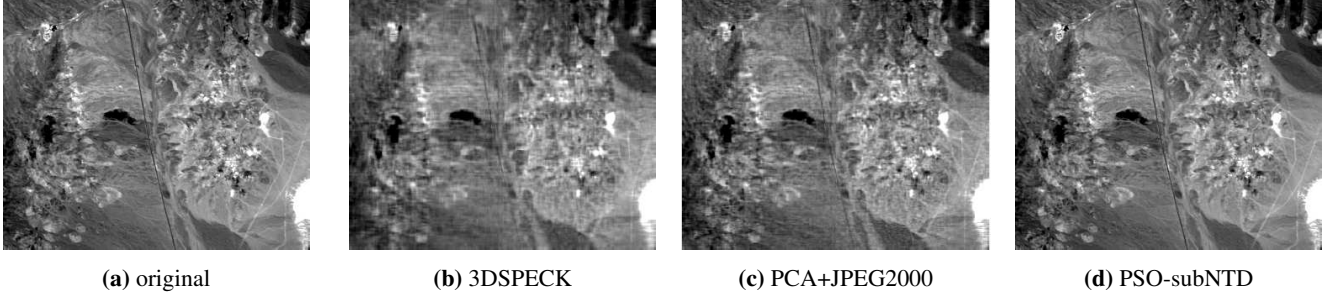


Fig. 3: Original and compressed band 170 of the reflectance Cuprite image at bpppb=0.2

Table 6: MSE for different compression methods on the reflectance Cuprite dataset

Method	0.05	0.1	0.2	0.5	1
3D-SPECK	1.1729	0.596	0.1978	0.1176	0.0645
PCA+JPEG2000	0.4270	0.3178	0.143	0.041	0.0294
PSO-NTD	0.2333	0.1444	0.0773	0.0174	0.011

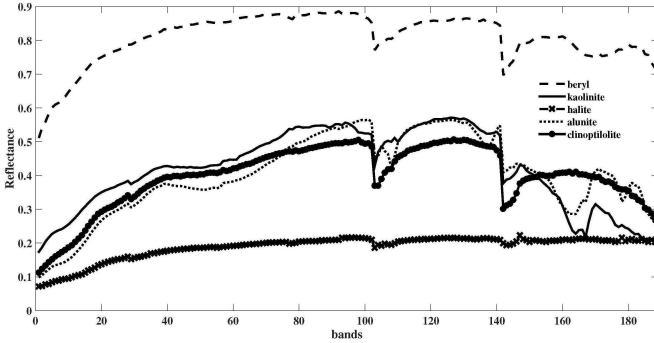


Fig. 4: Spectral signatures of five endmembers of the reflectance Cuprite dataset

In a second experiment, we apply the same experimental methodology to an urban dataset (Hydice). It has a size of 307×307 pixels and contains 210 bands. This dataset is available in [27]. The atmospheric and water absorption bands (bands [105-107],[140-151]) are removed and the remaining 195 bands are used. The Hydice dataset is partitioned into 25 subtensors (16 subtensors with size $64 \times 64 \times 195$, 4 subtensors with size $64 \times 51 \times 195$, 4 subtensors with size $51 \times 64 \times 195$ and one with size $51 \times 51 \times 195$) for (I'_1, I'_2, I'_3) .

Table 7 shows the SNR variance values at different bpppb ratios. Tables 8 shows the obtained MSE values. Again, PSO-subNTD outperforms the other methods, especially when the bpppb becomes lower than 0.5. Band 30 of the Hydice dataset at 0.2 bpppb is displayed in Fig.6. in Fig.7, the six most sig-

nificant endmembers of the Hydice dataset are plotted. The obtained abundance maps for these endmembers at 0.2 bpppb are displayed in Fig. 8. Similar conclusions as with the Cuprite image can be drawn.

When analysing more closely the result at bpppb=0.05 (CR= 320), we find that the compression ratio without arithmetic coding is 47.2. The extra compression ratios due to the arithmetic coding are $CR_{B^{*(1)}} = 4.88$, $CR_{B^{*(2)}} = 4.53$, $CR_{B^{*(3)}} = 2.1$ and $CR_{\underline{G}^*} = 10.86$ for the three matrices and the core tensor respectively. The CR of the endmember matrix \mathbf{E} and the endmember locations is 1.86 and 1.41 respectively.

Table 7: SNR variance (dB) for the different compression methods on the reflectance Hydice dataset

Method	0.05	0.1	0.25	0.5	1
3D-SPECK	12.11	14.02	18.39	23.41	30.34
PCA+JPEG2000	15.08 (PCs=15)	18.01 (PCs=20)	23.88 (PCs=30)	29.18 (PCs=50)	35.09 (PCs=100)
PSO-subNTD	19.07	22.3	27.15	32.23	36.91

Table 8: MSE for the different compression methods on the reflectance Hydice dataset

Method	0.05	0.1	0.2	0.5	1
3D-SPECK	3.4003	2.576	1.4568	0.3782	0.0734
PCA+JPEG2000	1.7243	1.341	0.424	0.0764	0.0181
PSO-subNTD	0.6448	0.4502	0.112	0.034	0.0078

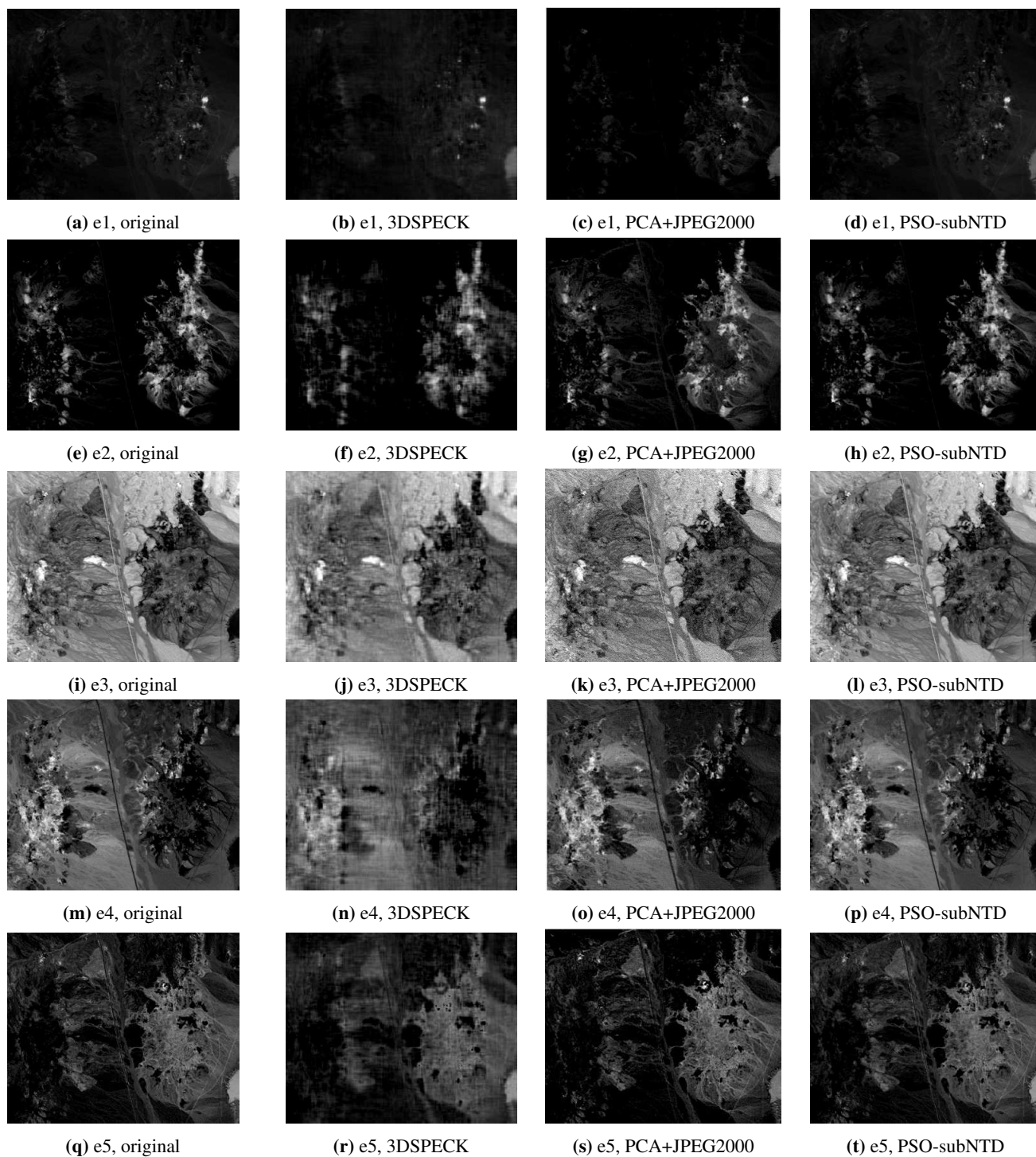


Fig. 5: Abundance maps for the five most significant endmembers of the reflectance Cuprite dataset: e1=beryl, e2=kaolinite, e3=halite, e4=alunite, e5=clinopillolite; original and after compression at bpppb=0.2

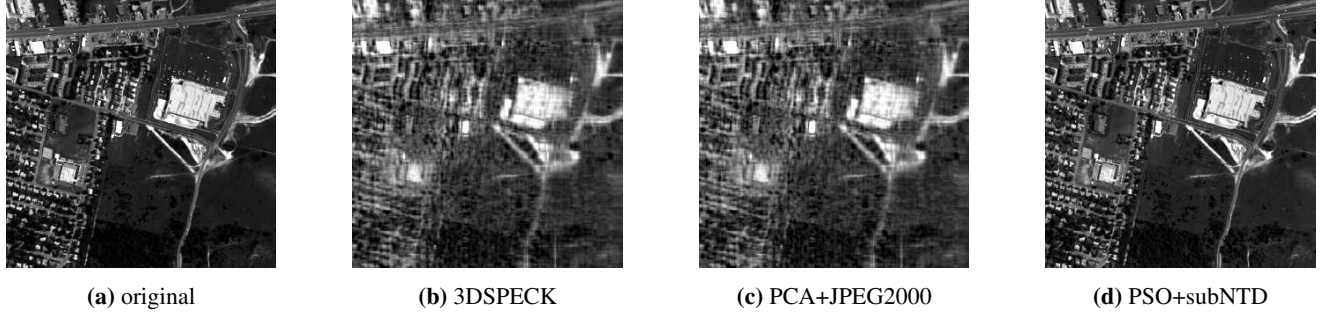


Fig. 6: Band 30 of the reflectance Hydice dataset; original and reconstructed at bpppb=0.2

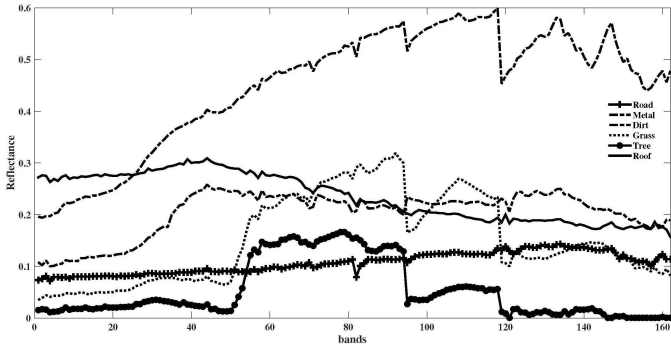


Fig. 7: Spectral signatures of six endmembers of the reflectance Hydice dataset

3.4. Fast subNTD

In the next experiment, the Fast subNTD method of section 2.5 is validated. In this case, specific core tensor dimensions are chosen, leading to a fixed compression ratio. The compression results for the synthetic and the two real reflectance datasets are shown in Tables 9-11. The fast subNTD can not be applied to radiance datasets because it requires the number of endmembers.

Table 9: Compression results on synthetic reflectance dataset, bpppb=1.5

dataset	SNR variance(dB)	MSE
3D SPECK	37.58	0.0524
PCA+ JPEG2000 (PCs=185)	39.08	0.0397
Fast subNTD	42.9	0.0237
PSO- subNTD	43.74	0.0208

Table 10: Compression results on Cuprite reflectance dataset, bpppb=1.5

dataset	ntering SNR variance(dB)	MSE
3D SPECK	41.34	0.057
PCA+ JPEG2000 (PCs=140)	42.62	0.0383
Fast subNTD	45.01	0.0096
PSO- subNTD	45.83	0.0084

Table 11: Compression results on Hydice reflectance dataset, bpppb=1.6

dataset	SNR variance(dB)	MSE
3D SPECK	35.51	0.0131
PCA+JPEG2000 (PCs=170)	39.28	0.0094
Fast subNTD	41.28	0.0067
PSO-subNTD	42.04	0.0053

In all cases, the fast version outperforms the other methods for a fixed compression ratio and gives results that are close to the ones from the optimal PSO-subNTD.

Based on the obtained results in this section, we can conclude that all compression methods show good performance at high bpppb values. For higher compression ratios, 3D-SPECK fails because it suffers from the difference in spectral versus spatial correlations of a HSI. PCA+JPEG2000 treats the spectral and spatial correlations separately and shows better results. The proposed method however obtains compression results that are significantly better. Moreover, the obtained spectral unmixing performances of the reconstructed HSI using PSO-subNTD are superior to those using the other methods.

We finalize with some words on the time complexity of the proposed method. The method is implemented in Matlab on a computer with an Intel(R) Core(TM) i7-4770 processor

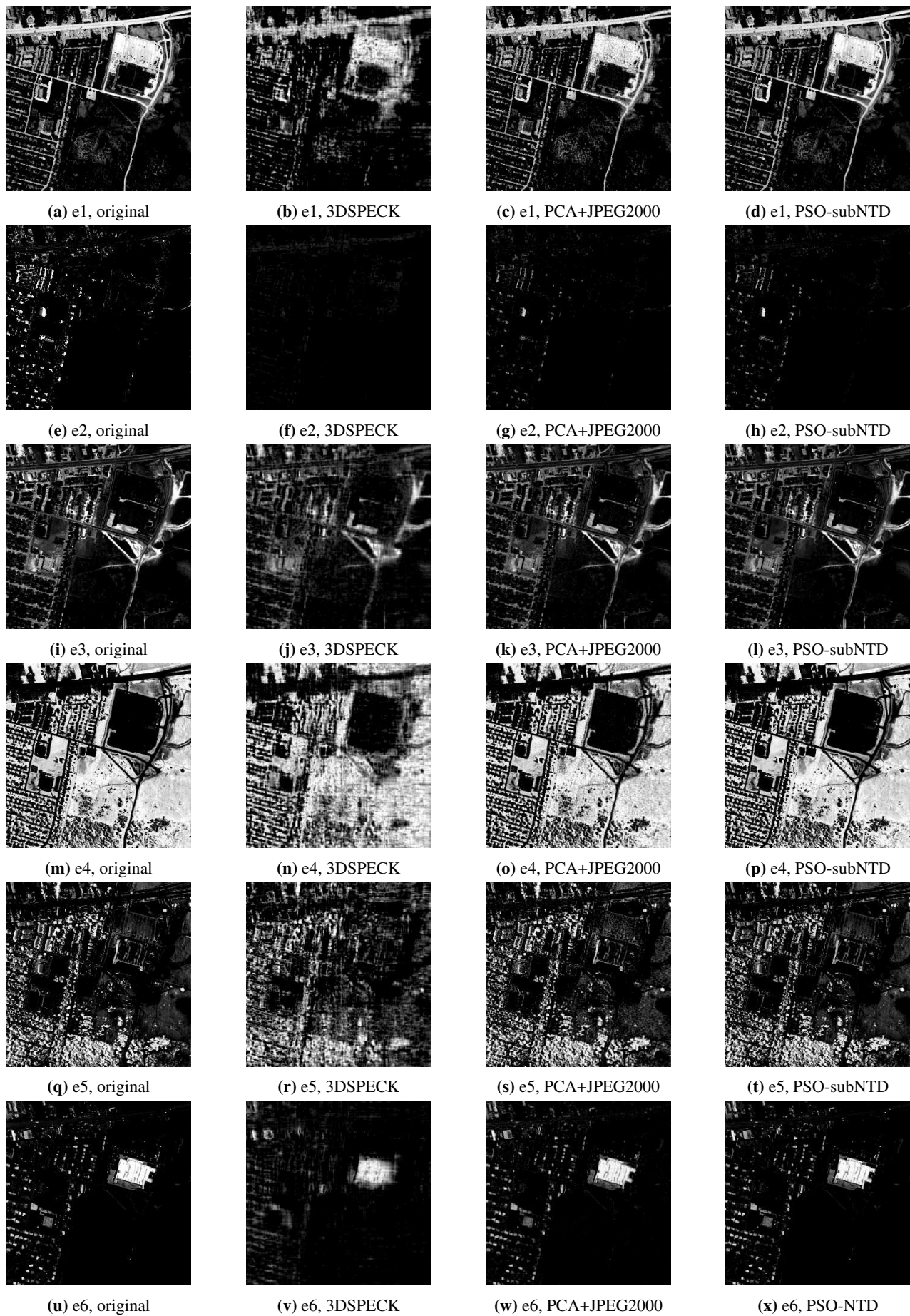


Fig. 8: Abundance maps for six endmembers of the reflectance Hydice dataset at bpppb=0.2; e1=Road, e2=Metal, e3=Dirt, e4=Grass, e5=Tree, e6=Roof

(3.9 GHz), 32GB of memory and a 64-bit Operating System. Since the method contains an iterative procedure for NTD and an optimization procedure (not in the case of the fast version), it is about 2 orders of magnitude slower than the optimized codes that were applied for the methods that we compare to. On the other hand, the applied procedures lend themselves easily to parallelization.

4. CONCLUSION

A new method for HSI compression using the Non-negative Tucker Decomposition is introduced. This method simultaneously minimizes the effect of compression on the spectral unmixing of the reconstructed image. The method determines the optimal reduced core tensor dimensions by minimizing the MSE between the abundance matrix of the original and reconstructed datasets by Particle Swarm Optimization.

The proposed algorithm achieves a better performance (higher SNR variance and smaller MSE) in comparison with two state-of-the-art compression algorithms, especially at high compression ratios. A fast approximate method that fixes the core tensor dimensions, is introduced as well. In future work, our aim is to reduce the computational complexity of the proposed algorithm.

ACKNOWLEDGMENT

The authors would like to thank Prof. James E. Fowler from Mississippi State University for sharing the QCCPACK and PCA+JPEG2000 toolboxes. Rob Heylen is a postdoctoral researcher sponsored by the Flemish fund for scientific research (FWO - Vlaanderen).

5. REFERENCES

- [1] J. Wang and C. Chang, "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1586–1600, Jun. 2006.
- [2] X. Tang and W. A. Pearlman, "Three-dimensional wavelet-based compression of hyperspectral images," in *Hyperspectral Data Compression*, G. Motta, F. Rizzo, and A. Storer, Eds. Norwell, MA: Kluwer Academic Publishers, 2006, ch. 10, pp. 273–308.
- [3] E. Christophe, C. Mailhes, and P. Duhamel, "Hyperspectral image compression: Adapting SPIHT and EZW to anisotropic 3-D wavelet coding," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2234–2346, Dec. 2008.
- [4] B. Penna, T. Tillo, E. Magli, and G. Olmo, "Progressive 3-D coding of hyperspectral images based on JPEG2000," *IEEE Geosci. and Remote Sens. Lett.*, vol. 3, no. 1, pp. 125–129, Jan. 2006.
- [5] Q. Du and J. E. Fowler, "Hyperspectral image compression using JPEG2000 and principal component analysis," *IEEE Geosci. and Remote Sens. Lett.*, vol. 4, no. 2, pp. 201–205, Apr. 2007.
- [6] A. Karami, M. Yazdi, and G. Mercier, "Compression and noise reduction of hyperspectral images using tucker decomposition and discrete wavelet transform," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 444–450, Mar. 2012.
- [7] M. Huber-Lerner, O. Hadar, S. R. Rotman, and R. Huber-Shalem, "Compression of hyperspectral images containing a subpixel target," *IEEE Trans. Geosci. Remote Sens.*, vol. 7, no. 6, pp. 2246–2255, 2014.
- [8] Q. Du, W. Zhu, and J. E. Fowler, "Anomaly-based JPEG2000 compression of hyperspectral imagery," *IEEE Geosci. and Remote Sens. Lett.*, vol. 5, no. 4, pp. 696–700, Oct. 2008.
- [9] B. Penna, T. Tillo, E. Magli, and G. Olmo, "Transform coding techniques for lossy hyperspectral data compression," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 5, pp. 1408–1421, May 2007.
- [10] F. G. Vlchez, J. M. Mar, M. Zortea, I. Blanes, V. G. Ruiz, G. C. Valls, A. Plaza, and J. S. Sagrista, "On the impact of lossy compression on hyperspectral image classification and unmixing," *IEEE Geosci. and Remote Sens. Lett.*, vol. 8, no. 2, pp. 253–257, Mar. 2011.
- [11] L. Santos, S. Lpez, G. M. Callic, J. F. Lpez, and R. Sarmiento, "Performance evaluation of the h.264/avc video coding standard for lossy hyperspectral image compression," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 451–461, Apr. 2012.
- [12] Q. Du, N. Ly, and J. E. Fowler, "An operational approach to PCA+JPEG2000 compression of hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2237–2245, Jun. 2014.
- [13] S. Sanchez and A. Plaza, "Real-time lossy compression of hyperspectral images using iterative error analysis on GPUs," in *Real-Time Image and Video Processing Conference*, 2012, pp. 84370G–1–9.
- [14] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Non-negative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley, Chichester, 2009.
- [15] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imaging Sci.*, vol. 6, no. 3, pp. 1758–1789, 2013.

- [16] Z. He, A. Cichocki, and S.Xie, "Efficient method for tucker3 model selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 15, pp. 805–806, 2009.
- [17] A. Huck and M. Guillaume, "Estimation of the hyperspectral tucker ranks," in *ICASSP Conference*, 2009, pp. 1281–1284.
- [18] J. Bioucas-Dias and J. Nascimento, "Hyperspectral subspace identification," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 8, pp. 2435–2445, Aug. 2008.
- [19] J. M. P. Nascimento and J. M. B. Dias, "Vertex component analysis: a fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, Apr. 2005.
- [20] R. Heylen, D. Burazerovic, and P. Scheunders, "Fully constrained least squares spectral unmixing by simplex projection," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4112–4122, Oct. 2011.
- [21] C. Ou and W. Lin, "Comparison between PSO and GA for parameters optimization of pid controller," in *IEEE International Conference on Mechatronics and Automation*, 2006, pp. 2471–2475.
- [22] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 6, pp. 58–73, Sep. 2002.
- [23] Z. Zhan, J. Zhang, Y. Li, and H. Chung, "Adaptive particle swarm optimization," *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 39, no. 6, p. 5873, 2009.
- [24] P. W. Jansen and R. E. Perez, "Constrained structural design optimization via a parallel augmented lagrangian particle swarm optimization approach," *Computers & Structures*, vol. 89, no. 13-14, pp. 1352–1366, Jul. 2011.
- [25] [Online]. Available: <http://speclab.cr.usgs.gov/spectral-lib.html> (Accessed 25 Sep. 2007).
- [26] [Online]. Available: <http://aviris.jpl.nasa.gov/html/aviris.freedata.html> (Accessed 15 Apr. 2015).
- [27] [Online]. Available: <http://www.tec.army.mil/hypercube> (Accessed 4 Sep. 2012).
- [28] [Online]. Available: <http://www.sandia.gov/~tgkolda/TensorToolbox/index-2.6.html> (Accessed 6 Feb. 2015).
- [29] B. Birge, "Particle swarm optimization toolbox." [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/7506-particle-swarm-optimization-toolbox>. (Accessed 20 Mar. 2006).
- [30] J. E. Fowler, "QccPack: an open-source software library for quantization, compression, and coding," in *Proc. SPIE Applications of Digital Image Processing XXIII*, A. Tescher, Ed., San Diego, CA., vol. 4115, 2000, pp. 294–301.
- [31] [Online]. Available: <http://www.kakadusoftware.com> (Accessed 20 Aug. 2014).
- [32] J. E. Fowler and J. T. Rucker, *3D wavelet-based compression of hyperspectral imagery*. John Wiley Sons, 2007, vol. 103, pp. 379–392.
- [33] A. B. Kiely and M. A. Klimesh, "Exploiting calibration-induced artifacts in lossless compression of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2672–2678, 2010.
- [34] C.-I. Chang and S. Wan, "Constrained band selection for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1575–1585, 2006.