Compression and noise reduction of hyperspectral images using non-negative tensor decomposition and compressed sensing

Shirin Hassanzadeh1* and Azam Karami2,3

1Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran
2Faculty of Physics, Shahid Bahonar University of Kerman, Kerman, Iran
3Visionlab, University of Antwerp, Belgium

*Corresponding author, e-mail address: shirin.hassanzadeh.e@gmail.com

Abstract
Hyperspectral images (HSI) are usually volumetric and require a lot of space and time for archiving and transmitting. In this research, a new lossy compression method for HSI is introduced based on non-negative Tucker decomposition (NTD). This method considers HSI as a 3D dataset: two spatial dimensions and one spectral dimension. The NTD algorithm decomposes the original data into a smaller 3D dataset (core tensor) and three matrices. In the proposed method, the Block Coordinate Descent (BCD) method is used to find the optimal decomposition, which is initialized by using Compressed Sensing (CS). The obtained optimal core tensor and matrices are coded by applying arithmetic coding and finally the compressed dataset is transmitted. The proposed method is applied to the real dataset, simulation results show that in comparison with well-known lossy compression methods such as 3D SPECK and PCA+JPEG2000, the proposed method achieves the highest signal to noise ratio (SNR) at any desired compression ratio (CR) while noise reduction is simultaneously acquired.

Keywords: Compression, hyperspectral images, non-negative Tucker decomposition, Compressed Sensing, Block Coordinate Descent, noise reduction.

Introduction
Hyperspectral images are used in a wide variety of applications such as agriculture, biomedical imaging, target detection, etc. HSI contain hundreds of bands that yield several gigabytes of data, which makes compression necessary to facilitate transmission and storage. There are two types of correlations in HSI which cause redundancy: spatial and spectral correlations. Therefore, compression can significantly reduce the hyperspectral data volumes. Different compression algorithms that have been recently proposed divide into two groups: lossy and lossless compression. In lossless compression, the obtained CR is around 3:1 which is not sufficient for many practical applications [Christophe, 2011; Su, 2014]. Recent lossy compression methods simultaneously reduce the spatial and spectral correlations, by considering HSI as...
a 3D dataset. The most popular of these methods are based on 3D wavelets such as the Set Partitioning in Hierarchical Trees (SPIHT) and Set Partitioning Embedded bloCK (SPECK) algorithms [Fowler and Rucker, 2007]. 3D SPECK has better performance than 3D SPIHT in terms of compression efficiency. Some other lossy compression methods separately reduce the spatial and spectral correlations. In Du and Fowler [2007], a PCA transform for spectral decorrelation in conjunction with JPEG2000 to provide spatial decorrelation was introduced. The results show that PCA has a better performance than the spectral Discrete Wavelet Transform (DWT). The Karhunen-love transform (KLT) also causes high spectral decorrelation, but has high computational complexity. A low complexity KLT-based algorithm in Blanes and Serra-Sagristà [2010] was introduced, its performance is better than DWT when it is used with the JPEG coding. In Cheng and Dill [2014], a lossless to lossy compression scheme for hyperspectral images based on a dual-tree Binary Embedded Zeroetree Wavelet (BEZW) algorithm has been presented. The impact of lossy compression on spectral unmixing, and supervised classification using Support Vector Machine (SVM) was investigated in García-Vílchez et al. [2011]. It was shown that for certain compression techniques, a higher compression ratio may result in more accurate classification. Another lossy compression algorithm based on PCA was presented in Huber-Lerner et al. [2014], the algorithm can enhance detection performance. A HSI compression method using preservation of bands of interest (so called BOI-preserving-based compression methods) was proposed in Chen et al. [2010]. Some bands of HSI are more important in some specific applications, and BOI selection methods can be chosen according to application requirements. BOI and non-BOI bands are respectively compressed with low distortion and high distortion. In Santos et al. [2012] it was demonstrated that the H.264/AVC video coding standard could be applied for the compression of hyperspectral cubes. It was shown that this method barely affected the accuracy of the endmember extraction. A 3D method of HSI compression based on NTD is presented in Karami et al. [2012]. In this method, a 2D-DWT is first applied to each spectral band of the HSI, after which NTD is applied to the wavelet subbands.

In this paper a new lossy compression method based on NTD is proposed. Popular existing NTD algorithms used least square methods [Kolda and Bader, 2009] which are not guaranteed to find the global optimal dimensions for the core tensor and the three matrices. In this paper we use a new algorithm based on BCD search [Xu and Yin, 2013] to find the optimal dimensions. Since BCD is sensitive to the initial values of the core tensor and the three matrices, we will employ a CS method [Caiafa and Cichocki, 2015] for the initialization. The obtained method (BCD-CS) is efficient and effective in finding the optimal solution, and simultaneously decreases the effect of Gaussian noise.

The rest of the paper is organized as follows: first the proposed method is presented. After that, the experimental results are shown and finally the concluding remarks are given.

Proposed Method
HSI are considered as a 3D tensor with two dimensions in the spatial and one dimension in the spectral domain. There are two types of correlation in HSI: spatial and spectral correlations. The spectral correlation is usually but not always stronger than spatial correlation (e.g. AVIRIS dataset). In the proposed method the non-negative tucker decomposition (NTD) is applied to the 3D HSI. It is explained in the following section.
**Block Coordinate Descent Search**

In this section a brief review of the Tucker model and BCD algorithm is presented. Important notations are shown in Table 1.

### Table 1 - Notations.

| Notation | Description |
|----------|-------------|
| $\mathbb{R}^n$ | n-dimensional real space vector |
| $Y$ | Third order tensor |
| $Y_{(n)}$ | n-mode matricization of tensor $Y$ |
| $A^{(n)}$ | n-mode matrix in Tucker model |
| $\times_n$ | n-mode product of a tensor by matrix |
| $\circ$ | Outer product |

A Tucker decomposition decomposes a third order tensor $X_{I_1 \times I_2 \times I_3}$ into a core tensor $W_{J_1 \times J_2 \times J_3}$ multiplied by three matrices (see Fig. 1) [Kolda and Bader, 2009]:

\[
X = \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \sum_{j_3=1}^{J_3} w_{j_1,j_2,j_3} a_{j_1} \circ a_{j_2} \circ a_{j_3} + E = \hat{X} + E
\]

where tensor $\hat{X}$ is an estimation of tensor $X$. Tensor $E$ denotes the estimation error. The quality of the estimation relies heavily on the dimensions of the core tensor. Our aim is to find the optimal components for the core tensor and the three matrices. Therefore the following optimization problem is considered:
Most NTD algorithms apply Alternative Least Squares (ALS) technique to solve this optimization problem. However, these methods do not guarantee to find the global optimal dimensions for the core tensor and the three matrices. Here we use the BCD algorithm for solving NTD [Xu and Yin, 2013]. It is proved this method could find the global optimal solution.

BCD is an iterative algorithm. In this algorithm $A_1, A_2, A_3$ and $W$ are considered as four variables. In this case, the optimization problem of Equation [2] is converted to:

$$\min \frac{1}{2} \|X - W \times_1 A_1 \times_2 A_2 \times_3 A_3\|_F^2$$

subject to:

- $W \in R^{I_1 \times J_2 \times J_3} \geq 0$
- $A_1 \in R^{I_1 \times J_1} \geq 0$
- $A_2 \in R^{I_2 \times J_2} \geq 0$
- $A_3 \in R^{I_3 \times J_3} \geq 0$  \[2\]

where $k$ is the number of iteration. In each iteration, one variable is updated while the other variables are fixed. This process is repeated until the convergence criterion is satisfied. The $W^k$ is updated as follows:

$$W^k = \max(0, \widetilde{W}^{k-1} - Grad^k W/P^k)$$  \[4\]

$$\widetilde{W}^k = W^k + \omega^k \ast (W^k - W^{k-1})$$  \[5\]
\[ \text{Grad}^k \mathbf{W} = [(\mathbf{W}^\ast)^{k-1} \times_1 (\mathbf{A}_1^{k-1})^T \mathbf{A}_1^{k-1} \times_2 (\mathbf{A}_2^{k-1})^T \mathbf{A}_2^{k-1} \times_3 (\mathbf{A}_3^{k-1})^T \mathbf{A}_3^{k-1}) -
\text{(Grad}^{k-1} \mathbf{W} \times_1 (\mathbf{A}_1^{k-1})^T \times_2 (\mathbf{A}_2^{k-1})^T \times_3 (\mathbf{A}_3^{k-1})^T)] \]  

[6]

\[ \omega^k = \min(\tau^k, r\sqrt{\frac{P^{k-1}}{P^k}}) \]  

[7]

\[ \tau^k = \frac{t^{k-1} - 1}{t^k} \]  

[8]

\[ t^k = 1 + \sqrt{1 + 4(t^{k-1})^2} \]  

[9]

\[ P^k = \| (\mathbf{A}_1^{k-1})^T \mathbf{A}_1^{k-1} \|_2 \times \| (\mathbf{A}_2^{k-1})^T \mathbf{A}_2^{k-1} \|_2 \times \| (\mathbf{A}_3^{k-1})^T \mathbf{A}_3^{k-1} \|_2 \]  

[10]

and \( \mathbf{A}_1^k, \mathbf{A}_2^k, \mathbf{A}_3^k \) are updated using:

\[ \mathbf{A}_n^k = \max(0, \widetilde{\mathbf{A}}_n^{k-1} - \text{Grad}^k \mathbf{A}_n^k / L_n^k) \]  

[11]

\[ \widetilde{\mathbf{A}}_n^k = \mathbf{A}_n^k + \tau_A^k \ast (\mathbf{A}_n^k - \mathbf{A}_n^{k-1}) \]  

[12]

\[ \text{Grad}^k \mathbf{A}_n^k = \widetilde{\mathbf{A}}_n^k (\mathbf{B}_n^k)^T (\mathbf{B}_n^k)^T - \mathbf{X}_n^k \]  

[13]

\[ L_n^k = \| (\mathbf{B}_n^k)^T (\mathbf{B}_n^k)^T \|_2 \]  

[14]

\[ \mathbf{B}_n^k = \mathbf{W}_n^k \times_n \mathbf{A}_n^{k-1} \]  

[15]

where \( \mathbf{B}_n^k \) is unfolding mode-n of \( \mathbf{B}_n^k \).
The required initial values are selected as follows:

\[ \mathbf{A}_1^0 = \mathbf{A}_1^0, \mathbf{A}_2^0 = \mathbf{A}_2^0, \mathbf{A}_3^0 = \mathbf{A}_3^0, \mathbf{W}^0 = \mathbf{W}^0, \text{Grad}^0 \mathbf{W} = \mathbf{X}, \]

\[ r = 1, \tau^0 = 1, \tau^0 = 1, L^0 = 1, P^0 = 1 \]

In the existing BCD algorithm, the initial values for the three matrices \( \mathbf{A}_1^0, \mathbf{A}_2^0, \mathbf{A}_3^0 \) and core tensor \( \mathbf{W}^0 \) are randomly generated. Since the existing algorithm is sensitive to the selected initial values, in the proposed method, appropriate initial values are obtained using CS. The CS method is briefly explained in the following section.

**Compressed Sensing**

Most algorithms introduced for solving NTD such as ALS [Kolda and Bader, 2009] and BCD [Xu and Yin, 2013] use random initial values for the three matrices and core tensor. In Kolda and Bader [2009], optimal initial values are introduced based on higher order singular value decomposition (HOSVD). However, HOSVD is a highly computationally expensive method which is also data dependent. For the first time, we use the CS method in order to estimate the initial values for three matrices and core tensor. In the CS method, first three sensing matrices are defined, these matrices are not dependent on the original dataset and could provide appropriate estimations of \( \mathbf{A}_1^0, \mathbf{A}_2^0, \mathbf{A}_3^0 \) and \( \mathbf{W}^0 \). In Caiafa and Cichocki [2015], different sensing matrices such as Gaussian and Bernoulli are introduced. It is shown that the Gaussian matrix has better efficiency in the reconstructed dataset. In our work, three sensing matrices \( \Phi_1, \Phi_2 \) and \( \Phi_3 \) are used in order to calculate three compressive measurements \( Z^{(1)}, Z^{(2)} \) and \( Z^{(3)} \) (see Fig. 2).

The compressive measurements can be computed as follows:

\[ Z^{(1)} = \mathbf{X} \times_2 \Phi_2 \times_3 \Phi_3 \]
\[ Z^{(2)} = \mathbf{X} \times_1 \Phi_1 \times_3 \Phi_3 \]  \[17\]
\[ Z^{(3)} = \mathbf{X} \times_1 \Phi_1 \times_2 \Phi_2 \]

The initial core tensor which is required for the BCD algorithm is obtained as follows:

\[ \overline{\mathbf{W}}^0 = \mathbf{X} \times_1 \Phi_1 \times_2 \Phi_2 \times_3 \Phi_3 \]  \[18\]

Three initial matrices are also calculated as:

\[ \mathbf{B}^k = \mathbf{W}^k \times_{-n} \{ \mathbf{A}^k \} \]  \[16\]
\[
\begin{align*}
A_n^0 &= Z_n W_n^\dagger \\
n &= 1, 2, 3
\end{align*}
\]  

where \( Z_n \) and \( W_n \) are unfolding matrices in mode-\( n \) of \( Z \) and \( W \) tensors (\( \dagger \) is the pseudo-inverse of a matrix).

CS is not an iterative technique, it is fast and can provide proper initial values for the core tensor and the three matrices. But existing tucker decomposition algorithms are usually based on using the random initial matrices and core tensor. If a good estimation for them could be considered the results of BCD will be better.

---

**Figure 2 - Three-way Compressive Measurements of a 3D tensor.**

**BCD-CS Algorithm**

In the proposed method, we apply CS to the original dataset to obtain appropriate initial values of the three matrices and core tensor, after which the BCD algorithm is used for solving the NTD. The optimum core tensor \( (\Phi) \) and matrices \( (A_1^0, A_2^0, A_3^0) \) obtained from BCD should be transmitted. Most elements of core tensor and three matrices are nearly zero. So we calculated the energy of coefficients of core tensor and then we preserved the elements representing 99.5% of the energy in HSI and discarded the other coefficients. We can also encode the elements in decreasing order, in several passes. For every pass a threshold can be chosen in a similar way to the bit plane encoding which all the elements are measured against threshold [Thyagarajan, 2011]. If an element is larger than the threshold it is encoded and removed from the core tensor and matrices; however if it is smaller, it is preserved for the next pass. When all the elements have been visited, the threshold is lowered and the tensor is scanned again to add more details to the already encoded...
tensor. This process is repeated until the energy of encoded elements equal to or larger than 99.5% of that of the original core tensor. The selected elements and their positions are then transferred.

In the following the proposed method is briefly summarized:

1. The original tensor is multiplied by three Gaussian sensing matrices respectively in mode-1, 2 and mode-3 to obtain three compressive measurement \( \mathbf{Z}^{(1)}(I_1 \times J_2 \times J_3) \), \( \mathbf{Z}^{(2)}(I_1 \times J_2 \times J_3) \), \( \mathbf{Z}^{(3)}(I_1 \times J_2 \times J_3) \) (see Eq. [17]).
2. The initial core tensor \( \mathbf{W}^{0}(I_1 \times J_2 \times J_3) \) is obtained from Equation [18].
3. The three initial matrices \( \mathbf{A}^0_1, \mathbf{A}^0_2, \mathbf{A}^0_3 \) are calculated using Equation [19].
4. The obtained \( \mathbf{A}^0_1, \mathbf{A}^0_2, \mathbf{A}^0_3 \) and \( \mathbf{W}^0 \) are used in BCD as initial values for the three matrices and the core tensor respectively.
5. The optimal values of matrices \( (\mathbf{A}^*, \mathbf{A}^*_2, \mathbf{A}^*_3) \) and core tensor \( (\mathbf{W}^*) \) are obtained by BCD.
6. Apply arithmetic coding to \( (\mathbf{A}^0_1, \mathbf{A}^0_2, \mathbf{A}^0_3) \) and \( \mathbf{W}^0 \).
7. Transmit compressed elements.
8. Reconstruct dataset \( \hat{\mathbf{X}} \).

The CR is defined as the total number of bits which is required for transmitting the original dataset \( (I_1 \times J_2 \times J_3) \) divided by the number of bits of the compressed dataset after applying arithmetic coding to the core tensor and matrices. In this way, CR is directly dependent on the core tensor size \( (J_1, J_2, J_3) \). The smaller values for \( (J_1, J_2, J_3) \), reach the higher CR and for each desired CR, proper values are manually selected.

**Experimental Results**

To evaluate the image quality, the signal-to-noise ratio (SNR) can be used. It estimates the quality of the reconstructed image \( \hat{\mathbf{X}} \) in comparison with the original one \( \mathbf{X} \). The SNR in dB is defined as:

\[
\text{SNR}_{\text{dB}} = 10 \log_{10} \left( \frac{\| \mathbf{X} \|^2}{\| \mathbf{X} - \hat{\mathbf{X}} \|^2} \right) \tag{20}
\]

\[
\| \mathbf{X} - \hat{\mathbf{X}} \|^2 = \sum_{k=1}^{I_1} \sum_{j=1}^{I_2} \sum_{i=1}^{I_3} (x_{ijk} - \hat{x}_{ijk})^2 \tag{21}
\]

where \( (I_1, I_2, I_3) \) is the original image size. Instead of CR, we employ the bitrate, defined as the number of bits per pixel per band (bpppb), which gives the average number of bits to represent a single pixel of the hyperspectral dataset.

In our experiments, a popular AVIRIS radiance dataset (Cuprite - http://aviris.jpl.nasa.gov/html/aviris.freedata.html) is used. This 16-bit dataset has been spatially cropped to a size 512×512 and it is composed of 224 spectral bands. Before applying the proposed method, the original dataset is partitioned into 64 patches (64×64×224). The proposed method is
compared to two state-of-the-art algorithms: PCA+JPEG2000 [Du and Fowler, 2007] and 3D-SPECK [Fowler and Rucker, 2007]. In these algorithms biorthogonal 9/7 wavelets are used. Since in HSI, the spectral correlation is usually higher than the spatial correlation, the wavelet decomposition level for the spectral domain should be chosen higher than for the spatial domain. After experimenting on the Cuprite data, we selected 6 spatial and 8 spectral levels for the wavelet decomposition in the 3D-SPECK algorithm. 3D-SPECK is implemented using the QccPack toolbox [Fowler, 2000]. The JPEG2000 coding is done using the Kakadu software (http://www.kakadusoftware.com) with a quantization step size of $10^{-7}$. For the proposed method, the maximum iteration number for BCD is selected as 5000. Figure 3 shows reconstructed images (15th band of the radiance cuprite dataset) using different compression methods.

![Figure 3 - Original and compressed band 15 of the radiance Cuprite image at bpppb=0.1.](image)

Figure 4 depicts the compression results for the Cuprite dataset. In order to indicate the efficiency of CS in the proposed method, the results of BCD without the CS step (with initial random values for core tensor and three matrices) are also shown in Figure 3. In comparison with the other techniques, BCD-CS has the highest SNR values at different bpppb.

In our experiment, we also investigated the impact of the proposed algorithm on noise reduction. The radiance values of the cropped Cuprite dataset are in the interval $[0, 11821]$.
then Gaussian noise with varying variances $5 \leq \sigma \leq 250$ were added to each band of the original dataset. Figure 5 shows the obtained SNR versus different noise variances for the noisy and reconstructed Cuprite dataset. Denoising results show that the proposed method improves SNR values with at least 5 dB. In fact, NTD is a higher-order form of Singular Value Decomposition (SVD). Since smaller singular values are usually corrupted by noise, by selecting $(J_1,J_2,J_3)$ smaller than the size of the original dataset $(I_1,I_2,I_3)$, the most noisy singular values are removed.

![Figure 4 - SNR versus bpppb for compressed images.](image)

![Figure 5 - SNR versus $\sigma$ for noisy and denoised Cuprite image with BCD-CS.](image)

**Conclusions**

In this paper, a new lossy compression method for HSI based on Block Coordinate Descent and Compressed Sensing algorithms is introduced. The proposed method is applied to the
real HSI. The results show that:

1) The reconstructed image achieves the highest SNR in comparison with two state-of-the-art compression algorithms (PCA+JPEG2000 and 3D-SPECK).

2) The proposed method also reduces the effect of Gaussian noise.

In future work, our aim is to reduce the computational complexity of the proposed algorithm, the method could automatically find the optimum values for core tensor size.

References

Blanes I., Serra-Sagristà J. (2010) - *Cost and scalability improvements to the Karhunen-Loève transform for remote-sensing image coding*. IEEE Transactions on Geoscience and Remote Sensing, 48 (7): 2854-2863. doi: http://dx.doi.org/10.1109/TGRS.2010.2042063.

Christophe E. (2011) - *Hyperspectral data compression tradeoff*. In: Optical Remote Sensing, 3: 9-29. doi: http://dx.doi.org/10.1007/978-3-642-14212-3_2.

Cheng K.J., Dill J. (2014) - *Lossless to lossy dual-tree BEZW compression for hyperspectral images*. IEEE Transactions on Geoscience and Remote Sensing, 52 (9): 5765-5770. doi: http://dx.doi.org/10.1109/TGRS.2013.2292366.

Chen H., Zhang Y., Zhang J., Chen Y. (2010) - *A BOI-preserving-based compression method for hyperspectral images*. IEEE Transactions on Geoscience and Remote Sensing, 48 (11): 3913-39230. doi: http://dx.doi.org/10.1109/tgrs.2010.2070511.

Caiafa C.F., Cichocki A. (2015) - *Stable, robust, and super fast reconstruction of tensors using multi-way projections*. IEEE Transactions on Signal Processing, 63 (3): 780-793. doi: http://dx.doi.org/10.1109/TSP.2014.2385040.

Du Q., Fowler J.E. (2007) - *Hyperspectral image compression using JPEG2000 and principal component analysis*. IEEE Transactions on Geoscience and Remote Sensing, 4 (2): 201-205. doi: http://dx.doi.org/10.1109/LGRS.2006.888109.

Fowler J.E., Rucker J.T. (2007) - *Three-dimensional wavelet-based compression of hyperspectral imagery*. In: Hyperspectral Data Exploitation: Theory and Applications, Chang C.-I. (Ed.), pp. 379-407. doi: http://dx.doi.org/10.1002/9780470124628.ch14.

Fowler J.E. (2000) - *QccPack: An open-source software library for quantization, compression, and coding*. In: International Symposium on Optical Science and Technology, International Society for Optics and Photonics, pp. 294-301.

García-Vílchez F., Muñoz-Mari J., Zortea M., Blanes I., González-Ruiz V., Camps-Valls G., Plaza A., Serra-Sagristà J. (2011) - *On the impact of lossy compression on hyperspectral image classification and unmixing*. IEEE Geoscience and Remote Sensing Letters, 8 (2): 253-257. doi: http://dx.doi.org/10.1109/LGRS.2010.2062484.

Huber-Lerner M., Hadar O., Rotman SR., Huber-Shalem R. (2014) - *Compression of Hyperspectral Images Containing a Subpixel Target*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7 (6): 2246-2255. doi: http://dx.doi.org/10.1109/JSTARS.2014.2320754.

Karami A., Yazdi M., Mercier G. (2012) - *Compression of hyperspectral images using discrete wavelet transform and tucker decomposition*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 5 (2): 444-450. doi: http://dx.doi.org/10.1109/JSTARS.2012.2189200.
Kolda T.G., Bader B.W. (2009) - *Tensor decompositions and applications*. SIAM review, 51 (3): 455-500. doi: http://dx.doi.org/10.1137/07070111X.

Santos L., Lopez S., Callico G.M., Lopez J.F., Sarmiento R. (2012) - *Performance evaluation of the H. 264/AVC video coding standard for lossy hyperspectral image compression*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 5 (2): 451-461. doi: http://dx.doi.org/10.1109/JSTARS.2011.2173906.

Su H, Du P. (2014) - *Multiple Classifier Ensembles with Band Clustering for Hyperspectral Image Classification*. European Journal of Remote Sensing, 47: 217-227. doi: http://dx.doi.org/10.5721/EuJRS20144714.

Thyagarajan K.S. (2011) - *Still Image and video compression with MATLAB*. John Wiley & Sons.

Xu Y., Yin W. (2013) - *A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion*. SIAM Journal on Imaging Sciences, 6 (3): 1758-1789. doi: http://dx.doi.org/10.1137/120887795.

© 2016 by the authors; licensee Italian Society of Remote Sensing (AIT). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).