

HYPERSPPECTRAL IMAGE CODING USING GRAPH WAVELETS

Jin Zeng^{*}, Gene Cheung[#], Yung-Hsuan Chao[%], Ian Blanes[§], Joan Serra-Sagristà[§], Antonio Ortega[%]

^{*} The Hong Kong University of Science and Technology, [#] National Institute of Informatics,
[%] University of Southern California, [§] Universitat Autònoma de Barcelona

ABSTRACT

Hyperspectral imaging captures the spectral responses of different wavelengths per pixel for an entire image. Because the number of spectral bands is large, efficient compression of hyperspectral images is important. Leveraging on recent advances in graph signal processing (GSP), in this paper we propose to encode a hyperspectral image in groups of ω spectral bands using graph wavelets, exploiting correlations along both the spatial and the spectral dimensions. Specifically, along the spatial dimension, we estimate the inter-pixel correlations for all adjacent pixel pairs from the last image in the previous coded band group. Along the spectral dimension, we first divide an image into different spatial regions with similar spectral responses, and encode the spectral signature (correlations along the spectrum) for each region as side information (SI). The spatial / spectral correlations are used to compute edge weights to construct a graph for signal-adaptive graph wavelet based compression. Experimental results suggest that our proposal can outperform existing schemes noticeably at comparable complexity.

Index Terms— hyperspectral imaging, image compression, graph signal processing

1. INTRODUCTION

A hyperspectral image contains spectral responses of different wavelengths for each pixel in an image, resulting in a large volumetric dataset. Due to its large size, compression of hyperspectral images is important for storage and transmission. Previous transforms proposed for lossy hyperspectral image coding [1–7] can be broadly divided into two categories. In the first category are fixed transforms such as the Discrete Wavelet Transform (DWT), which do not adapt transform basis to the nature of the input signal, but can be efficiently implemented in low complexity [1, 2]. In the second category are signal-adaptive transforms such as the Karhunen-Loève Transform (KLT), which adjust its transform basis according to the input signal towards sparse signal representation, but typically suffer from high computation complexity [3, 4].

In this paper, we optimize the usage of a critically sampled biorthogonal graph wavelet design called *graphBior* [8] for hyperspectral image compression. Unlike fixed transforms, *graphBior* is signal-adaptive and leads to better coding performance in general, yet unlike KLT, complexity of *graphBior* is on par with that of DWT. *To the best of our knowledge, we are the first to employ signal-adaptive graph wavelets for hyperspectral image compression.* The key to successful deployment of *graphBior* is the proper design of an underlying graph with edge weights reflecting inter-node correlations.

Along the spatial dimension, we partition spectral bands into groups of ω consecutive bands for separate coding (for random access purpose), and estimate inter-pixel correlations for all adjacent pixel pairs from the last image in the previous coded band group. Inter-pixel correlations are then mapped to corresponding edge weights in the spatial sub-graph.

Along the spectral dimension, we divide an image into different spatial regions with similar spectral responses, and encode the *spectral signature* (correlations along the spectrum) for each region as side information (SI). The encoded spectral signatures are mapped to edge weights in the spectral sub-graph. In a low-complexity mode, we use *graphBior* to perform filtering in the spatial and spectral sub-graphs in sequence. In a high-complexity mode, we use *graphBior* to perform filtering in the spatial sub-graph and use KLT along the spectral dimension. Experimental results show that using signal graph wavelets can lead to coding gain for both low- and high-complexity modes. Note that we can apply other filter designs besides *graphBior* with the same approach for graph construction as proposed.

The outline of the paper is as follows. We first discuss related works in Section 2. We provide an overview of graph wavelets in Section 3 and describe our coding strategy in Section 4. Finally, experiments and conclusion are presented in Section 5 and 6, respectively.

2. RELATED WORK

Given that correlation among spectral components is more significant than correlation among spatial samples, lossy encoding of hyperspectral images benefits from a separable approach, where a 1D transform is first applied along the spectral dimension and then a 2D transform is applied in the spa-

UAB was supported by Spanish Ministry of Economy and Competitiveness and European Regional Development Fund. Grant TIN2015-71126-R.

tial dimension. Coding techniques providing the highest rate-distortion performance apply a KLT in the spectral dimension and are then coupled with JPEG 2000 Part 1 [9], as in [3, 4]. Recent contributions aim at providing equivalent coding performance to KLT at a lower computational cost, either by applying a spatially subsampled KLT [3], a low complexity KLT [10], a spectrally clustered KLT [6, 7], or by specially crafted transforms [11]. Here we propose an innovative, efficient and computationally feasible approach based on new coding tools in graph signal processing.

3. OVERVIEW OF GRAPH WAVELETS

Consider an undirected, weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ composed of a vertex set \mathcal{V} of size N and an edge set \mathcal{E} specified by $(i, j, w_{i,j})$, where $i, j \in \mathcal{V}$, and $w_{i,j} \in \mathbb{R}^+$ is the edge weight between vertices i and j . Thus a weighted graph can be characterized by its adjacency matrix \mathbf{W} with $\mathbf{W}(i, j) = w_{i,j}$. A graph-signal is a mapping that assigns a value to each vertex, denoted as $\mathbf{x} = [x_1, \dots, x_N]^T$.

A bipartite graph is a graph whose vertices can be divided into two disjoint sets such that no two vertices within the same set are adjacent. To compactly represent a graph-signal in the frequency domain, critically sampled biorthogonal wavelet filterbanks (graphBior) [8] have been proposed to decompose signals on bipartite graphs into low-pass and high-pass components. GraphBior filterbanks lead to compact signal representation in the wavelet domain, resulting in good compression performance. However, graph-signals typically live on general graphs that are not bipartite, so the original non-bipartite graph must be decomposed into a sequence of bipartite subgraphs, which can be accomplished with methods such as [12, 13]. Then the filterbanks can be applied iteratively in each subgraph, similar to separable filter for images where filters in x - and y -dimension are applied separately.

4. GRAPH-BASED CODING FOR HYPERSPECTRAL IMAGING

We now discuss how we construct a sparse graph for a hyperspectral image, so that graphBior can exploit its spatial and spectral correlation for efficient compression. Let \mathbf{x} be a given hyperspectral image with K spectral bands, each of size M . $x_{i,k}$ denotes the intensity value of the i -th pixel in the k -th band, where $i \in \{1, \dots, M\}$ and $k \in \{1, \dots, K\}$. The graph for signal \mathbf{x} is denoted by $\mathcal{G}_{\mathbf{x}}(\mathcal{V}, \mathcal{E})$, where each vertex represents one pixel, and links between vertices can be spatial (\mathcal{S}) or spectral (\mathcal{F}) so that $\mathcal{S} \cup \mathcal{F} = \mathcal{E}$, $\mathcal{S} \cap \mathcal{F} = \emptyset$.

4.1. Spatial Domain Connection

In hyperspectral images, spatial correlations are similar across spectral bands. Hence once a spatial graph structure is learned in one band, it can be reused in the subsequent

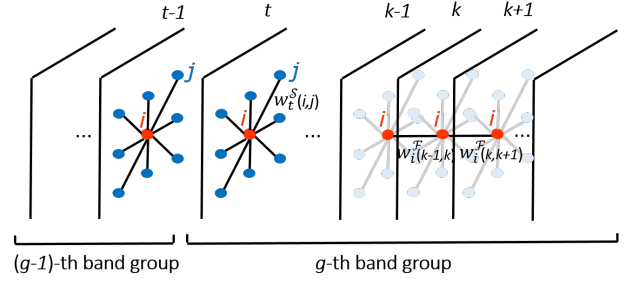


Fig. 1. Graph representation of hyperspectral image.

bands. To facilitate random access (component scalability), we first divide the K spectral bands into groups of ω consecutive bands. Then, for bands in the g -th group, the weights of spatial edges are learned from the last image in the previously coded $(g-1)$ -th group. For example, as shown in Fig. 1, in the first band of the g -th group (denoted as the t -th band), each vertex is connected to its eight spatial neighbors of the same band, since pixel values are typically correlated locally.

If there exists a sharp change in pixel value, the inter-pixel correlation is weak. The weight of an edge is thus computed based on intensity difference using a Gaussian kernel. For instance, the edge weight $w_t^S(i, j)$ between vertex $v_{i,t}$ and its neighbor $v_{j,t}$ is computed as:

$$w_t^S(i, j) = \exp\left(-\frac{(\hat{x}_{i,t-1} - \hat{x}_{j,t-1})^2}{\sigma^2}\right), \quad (1)$$

where $t-1$ is the index of the last band in the $(g-1)$ -th group, $\hat{x}_{i,t-1}, \hat{x}_{j,t-1}$ are the decoded values, and σ is a parameter to adjust weight sensitivity to intensity difference. The function in (1) is widely used to compute graph weights for various applications [14, 15]. Then the following bands in the g -th group will reuse the same graph in the t -th band.

4.1.1. Low Complexity Implementation

To reduce computation complexity, instead of using a Gaussian kernel to compute each edge weight using (1), we use the inter-pixel difference value to select one discrete value from a pre-determined set via thresholding. Specifically, given the number of quantization bins, we first determine the bin boundaries for the Gaussian function in (1) using the Lloyd's algorithm [16]. We can then determine the correct bin a given inter-pixel difference belongs to via a binary decision tree of thresholds (bin boundaries). Given the correct bin, the edge weight for the inter-pixel difference is the bin centroid. The bin boundaries are optimized once for all images, and as part of the codec, there is no side information overhead per image.

4.1.2. Executing graphBior Spatially

Given the constructed spatial-domain edge connections, we can apply graphBior along the spatial dimension. However,

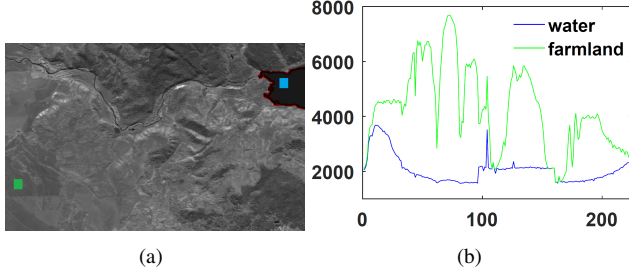


Fig. 2. (a) A patch of band 50 in Yellowstone uncalibrated scene0 with water pixel in blue and farmland pixel in green, and contour of water region marked in red; (b) Spectral signatures for water and farmland pixel.

the spatial graph is four-colorable, and hence the graph must first be decomposed into two bipartite subgraphs (indexed by l) for two successive executions of graphBior. For simplicity, we allocate horizontal and vertical edges into one bipartite subgraph ($l = 1$), and diagonal and anti-diagonal edges into the other subgraph ($l = 2$). Then with 2-dimensional graphBior, the signal is decomposed into four channels, *i.e.*, $\{LL, LH, HL, HH\}$ channels, where sparsity is achieved in high-pass channels.

To enable multi-level decomposition, the wavelet coefficients are downsampled to LL channel after one level decomposition, and for each bipartite subgraph, the adjacency matrix $\mathbf{W}_{t,l}^{S'}$ for vertices in LL of t -th band is given by the square of the current adjacency matrix $\mathbf{W}_{t,l}^S$, *i.e.*, $\mathbf{W}_{t,l}^{S'} = (\mathbf{W}_{t,l}^S)^2(LL, LL)$. Horizontal/Vertical edges are removed from $\mathbf{W}_{t,2}^{S'}$ so that the resulting downsampled sub-graph only contains diagonal edges and remains bipartite.

4.2. Spectral Domain Connection

It has been observed that spectral domain correlations are distinctly different for different materials [17]. For example, Fig. 2 shows the *spectral signatures* (intensities along the spectral dimension) for two pixels in water and farmland respectively. The Pearson correlation coefficient [18] between the two signatures is as low as -0.07, which confirms their differences. Therefore, it is reasonable to have different edge structures for regions with different materials.

In practice, for our specific class of test images, we only segment out water regions (if they exist), since water has a very unique spectral signature, and is easy to detect via thresholding due to its low frequency response compared with other materials. Specifically, for each image, we examine band 50, and if the pixel intensity is less than a threshold (pre-set to 2200), it is marked as a white pixel; otherwise it is marked as a black pixel. Then we apply Canny edge detection [19] to find the contour of the water region, as shown in Fig. 2(a), then encode it using arithmetic edge coding (AEC) [20], which has been shown to be efficient.

After segmentation, we learn the edge weights for adjacent bands for different regions based on their spectral signatures. Specifically, we calculate the average spectral vectors $\{\beta_p\}_{p=1,2}$ for pixels in ($p = 1$) or outside ($p = 2$) the water region. Then the edge weights are computed based on the inter-pixel differences. For example, as shown in Fig. 1, a pixel in the k -th band is linked to pixels of the same spatial location in adjacent bands, resulting in a line graph. The edge weight, for example the one between $v_{i,k}$ and $v_{i,k+1}$ in region p is computed as:

$$w_i^{\mathcal{F}}(k, k+1) = \exp\left(-\frac{(\beta_p(k+1) - \beta_p(k))^2}{\sigma_p^2}\right), \quad (2)$$

where σ_p^2 is the variance of $\{|\beta_p(k+1) - \beta_p(k)|\}_{k=1}^{K-1}$.

4.2.1. Low Complexity Implementation

Similar to the quantization procedure in Sec. 4.1, we replace the Gaussian kernel in (2) with a set of discrete values to reduce computational cost. The resulting spectral weights are encoded with arithmetic coding [21] to be sent as side information. In addition, to enable multi-level decomposition, after one level decomposition, the wavelet coefficients are downsampled to L channel and the adjacency matrix $\mathbf{W}_p^{\mathcal{F}'}$ for vertices in L within region p is given by the square of the current adjacency matrix $\mathbf{W}_p^{\mathcal{F}}$, *i.e.*, $\mathbf{W}_p^{\mathcal{F}'} = (\mathbf{W}_p^{\mathcal{F}})^2(L, L)$.

4.3. Coding Scheme for Low/High Complexity Mode

Under *low complexity mode*, we construct a graph and apply graphBior in both spectral and spatial domains. Filtering operations in graphBior is of complexity $\mathcal{O}(N)$ [8], which is on par with that of DWT. Given the hyperspectral image \mathbf{x} divided into band groups of size ω each, and quantization parameter (QP), the implementation takes the following steps:

1. Detect water region, and encode its contour with AEC;
2. Find the spectral signatures β_p , $p = 1, 2$, and encode the spectral weights with arithmetic coding;
3. For the first band group, apply multi-level spectral graphBior and spatial DWT, and encode the wavelet coefficients with AGP [22] for given QP.
4. For the following groups, first decode the last image in previously encoded group and learn the spatial graph; then apply multi-level graphBior in both spectral and spatial domain, and encode the wavelet coefficients with AGP for given QP.

Under *high complexity mode*, we replace spectral graphBior with KLT, which provides better spectral decorrelation:

1. For the first band group, apply multi-level spectral KLT and spatial DWT, and encode the wavelet coefficients with AGP for given QP;

2. For the following groups, first decode the last image in previous group and learn the spatial graph; then apply multi-level spectral KLT and spatial graphBior, and encode the wavelet coefficients with AGP for given QP.

5. EXPERIMENTAL RESULTS

In this section, we demonstrate the lossy coding performance of the proposed scheme in low-/high-complexity mode. We employ the datasets of AVIRIS Yellowstone uncalibrated scene 0, 3, 10, 11 and 18 [23], using 224 bands of size 512×680 . Each dataset is divided into 7 groups given $\omega = 32$.

5.1. Experiment Settings

In low-complexity mode, we apply 5-level graphBior decomposition in both spatial and spectral domain (denoted as graphBior in demonstration). To demonstrate the effect of spectral graphBior, we additionally generate results with spectral edges all set to 1 (denoted as graphBior w/o spectral edge). We compare to JPEG2000 multi-component compression [24], which has been exploited by several authors for hyperspectral image compression [25]. Irreversible 9/7 Integer Wavelet Transform (IWT) is applied with 5-level in spatial domain and 5-level in spectral domain, implemented with Kakadu software [26].

In high-complexity mode, we apply spectral KLT and 5-level spatial graphBior (KLT+graphBior). We compare to spectral KLT coupled with spatial DWT CDF 9/7 (KLT+DWT) which is reported to provide the best results in existing works [25]. The irreversible Karhunen Loève Transform is computed with the Spectral Transform software [27] and applied along the spectral dimension, and 5 levels of DWT are applied in the spatial dimension, then the wavelet coefficients are encoded with AGP.

5.2. Rate-Distortion Performance

The rate-distortion curves are shown in Fig. 3. In low-complexity mode, it is shown that our proposed graphBior-based coding scheme outperforms JPEG2000 significantly, and graphBior with spectral edge is better than graphBior without spectral edge. We conclude that our graph construction in spatial and spectral domains is effective.

In high-complexity mode, KLT+graphBior outperforms KLT+DWT, and the SNR gain at low bitrate is larger than at high bitrate, since the improvement due to spatial wavelet transform vanishes for medium to high bitrate. Note that the result here is not as good as the one with KLT coupled with JPEG2000 due to the adopted entropy coding scheme, but since we focus on comparing different transforms, it is fair to use the same entropy coding.

In Table 1, we show the average SNR gain of different schemes over JPEG2000, calculated with the Bjontegaard

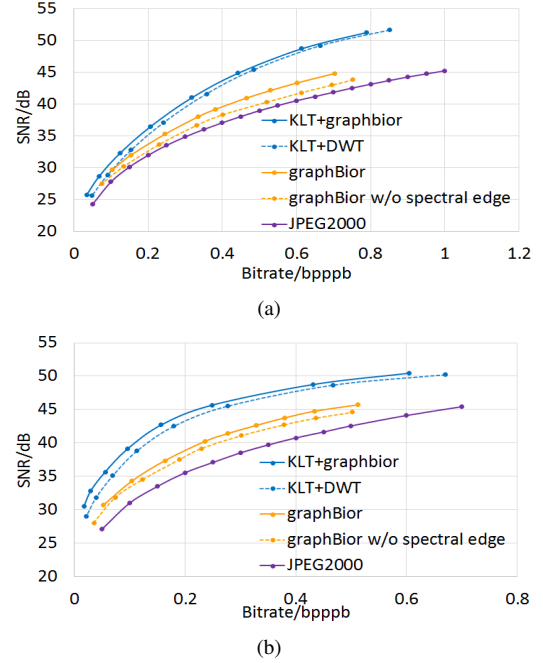


Fig. 3. R-D curve for different transforms. The images used are the Yellowstone uncalibrated (a) scene0 and (b) scene10.

metric [28]. High-complexity KLT+graphBior has about 1dB gain over KLT+DWT. Low-complexity graphBior has about 2 to 3dB gain over JPEG2000, and graphBior w/o spectral edge also has 1 to 2dB gain. This again validates the advantage of graph wavelet transform over fixed transform.

Table 1. Average gain of different schemes over JPEG2000 in SNR(dB).

	High complexity		Low complexity	
	KLT+graphBior	KLT+DWT	graphBior	w/o spectral
sce0	4.82	3.92	2.19	0.96
sce3	9.69	9.04	2.72	1.36
sce10	8.36	7.17	3.59	2.47
sce11	8.30	6.68	3.32	2.09
sce18	8.47	7.96	1.95	1.15

6. CONCLUSION

In this paper, we propose a graph-wavelet based hyperspectral image compression scheme. To enable signal adaptivity in the wavelet basis, we construct our graph based on spatial and spectral correlation. Specifically, along the spatial dimension, the edge weights for all adjacent pixel pairs are estimated from the last image in the previous coded band group. Along the spectral dimension, the image is divided into different spatial regions with similar spectral responses, and the spectral signature for each region is used to decide spectral edge weights. Experimental results suggest that our proposal can outperform existing schemes at comparable complexity.

7. REFERENCES

- [1] Xiaoli Tang and William A. Pearlman, "Three-dimensional wavelet-based compression of hyperspectral images," in *Hyperspectral Data Compression*, pp. 273–308. Springer, 2006.
- [2] James E. Fowler and Justin T. Rucker, "Three-dimensional wavelet-based compression of hyperspectral imagery," *Hyperspectral Data Exploitation: Theory and Applications*, pp. 379–407, 2007.
- [3] Barbara Penna, Tammam Tillo, Enrico Magli, and Gabriella Olmo, "Transform coding techniques for lossy hyperspectral data compression," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1408–1421, 2007.
- [4] Qian Du and James E. Fowler, "Hyperspectral image compression using JPEG2000 and principal component analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 2, pp. 201–205, 2007.
- [5] Jing Zhang, James E. Fowler, and Guizhong Liu, "Lossy-to-lossless compression of hyperspectral imagery using three-dimensional TCE and an integer KLT," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, pp. 814–818, Oct. 2008.
- [6] Ian Blanes and Joan Serra-Sagrà, "Cost and scalability improvements to the Karhunen-Loève transform for remote-sensing image coding," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 7, pp. 2854–2863, Jul. 2010.
- [7] Ian Blanes and Joan Serra-Sagrà, "Pairwise orthogonal transform for spectral image coding," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 3, pp. 961–972, Mar. 2011.
- [8] Sunil K. Narang and Antonio Ortega, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4673–4685, 2013.
- [9] "JPEG2000 image coding system part 1: Core coding system," in *ISO/IEC 15444-1*, 2000.
- [10] Jing Huang and Rihong Shu, "Hyperspectral image compression using low complexity integer KLT and three-dimensional asymmetric significance tree," *Proceedings of SPIE*, vol. 7444, Aug. 2009.
- [11] Naoufal Amrani, Joan Serra-Sagrà, Miguel Hernández-Cabronero, and Michael Marcellin, "Regression wavelet analysis for progressive-lossy-to-lossless coding of remote-sensing data," in *Data Compression Conference (DCC), 2016*. IEEE, 2016, pp. 121–130.
- [12] Sunil K. Narang and Antonio Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2786–2799, 2012.
- [13] Jin Zeng, Gene Cheung, and Antonio Ortega, "Bipartite subgraph decomposition for critically sampled wavelet filterbanks on arbitrary graphs," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 6210–6214.
- [14] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [15] Wei Hu, Gene Cheung, Antonio Ortega, and Oscar C Au, "Multiresolution graph fourier transform for compression of piecewise smooth images," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 419–433, 2015.
- [16] Stuart Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [17] Monika Stadnicka, Ian Blanes, Joan Serra-Sagrà, and Michael W Marcellin, "Mitigating discontinuities in segmented Karhunen-Loève Transforms," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2211–2215.
- [18] Karl Pearson, "Note on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895.
- [19] John Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , no. 6, pp. 679–698, 1986.
- [20] Ismael Daribo, Dinei Florencio, and Gene Cheung, "Arbitrarily shaped motion prediction for depth video compression using arithmetic edge coding," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4696–4708, 2014.
- [21] Khalid Sayood, *Introduction to data compression*, Newnes, 2012.
- [22] Amir Said and William A. Pearlman, "Low-complexity waveform coding via alphabet and sample-set partitioning," in *Information Theory. 1997. Proceedings. IEEE International Symposium on*. IEEE, 1997, p. 61.
- [23] NASA Jet Propulsion Laboratory, "Avisir hyperspectral images," <http://compression.jpl.nasa.gov/hyperspectral/>, 2010-04. Retrieved: 2017-01-15.
- [24] "JPEG2000 image coding system part 2: Extensions," in *ISO/IEC 15444-2*, 2004.
- [25] Ian Blanes, Enrico Magli, and Joan Serra-Sagrà, "A tutorial on image compression for optical space imaging systems," *IEEE Geoscience and Remote Sensing Magazine*, vol. 2, no. 3, pp. 8–26, 2014.
- [26] David Taubman, "Kakadu software," <http://kakadusoftware.com/>, 2017-01. Retrieved: 2017-01-05.
- [27] GICI-UAB, "Spectral transform software," <http://www.gici.uab.cat/GiciWebPage/downloads.php>, 2010-01. Retrieved: 2017-01-15.
- [28] Gisle Bjontegaard, "Calculation of average PSNR differences between RD-curves," *Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA, 2-4 April 2001*, 2001.