Compression of Hyperspectral Images With LVQ-SPECK

CIPR Technical Report TR-2008-3

Alessandro J. S. Dutra, William A. Pearlman, Eduardo A. B. da Silva

IEEE ISCAS May 2008

Center for
Image Processing Research

Rensselaer Polytechnic Institute
Troy, New York 12180-3590
http://www.cipr.rpi.edu

Hyperspectral Image Coding with LVQ-SPECK

Alessandro J. S. Dutra
ECSE Department
Rensselaer Polytechnic Institute
Troy, NY – 12180
a.dutra@ieee.org

William A. Pearlman
ECSE Department
Rensselaer Polytechnic Institute
Troy, NY – 12180
pearlman@ecse.rpi.edu

Eduardo A. B. da Silva
PEE-COPPE/DEL
Universidade Federal do Rio de Janeiro
Cx. P. 68504, Rio de Janeiro, RJ
Brasil, 21945-970
eduardo@lps.ufrj.br

***Abstract*—Enhanced versions of the LVQ-SPECK algorithm are presented that further exemplify the codec's good performance when dealing with multidimensional data sets. The two different alternatives might, in fact, be considered stepwise improvements over the original codec. First, an extended-range option is implemented, so that the number of spectral bands being simultaneously encoded is now a multiple of (instead of equal to) the codeword dimension. That in itself provides the means for much better rate allocation among the different bands. Then, use of the discrete wavelet transform over the spectral dimension, generating a wavelet packet decomposition of the original dataset is considered, and a substantial increase in coding performance is obtained, provided by the energy compaction characteristics of the transform. We provide results for the two different options of four-dimensional codebooks investigated, namely the shell-1 and shell-2 of the D_4 lattice, further cementing the idea that the latter is much more adept for use in this setting than the former. Finally we present comparison with other state-of-the-art codecs employed to the same task and show how competitive our proposed extensions are.**

I. Introduction

In a recent study [12] we presented a multidimensional image codec, derived from the state-of-the-art SPECK codec [2], that employed a lattice vector quantizer (vq) subset to the task of compressing a number of adjacent spectral bands. That new algorithm, called LVQ-SPECK, combined the well-established ideas of quadtree encoding present in SPECK with a successive-approximation process tailored to be used with vector-valued codewords which, in that case, are points extracted from selected lattices in a given dimension.

During the development of the LVQ-SPECK codec, particular attention was given to its performance in the encoding of hyperspectral images. Given the usually large amount of data involved in the acquisition of such images and the range of applications in which they are involved, this is still a considerably active area of research, with many different routes being followed. Methods spanning from direct quantization of spectral values [6] to those that employ the discrete wavelet transform [3] as a decorrelating step were developed, providing good compression capabilities along with good quality representation - even lossless, if desired.

In [6], Motta et al. define a partition of the spectral space whose boundaries are optimized by repeated application of a Generalized Lloyd Algorithm [9] (GLA) variant. Considering the original data set to have a dimension D, the design of a D-dimensional vector quantizer, which is usually computationally prohibitive, would be required. Instead, the method chooses to design N vector quantizers, each with dimension d_i, where $\sum_{i=0}^{N} d_i = D$. The resulting Partitioned Vector Quantizer is then the Cartesian product of all the lower dimensional dictionaries. In order to remove part of the remaining source redundancy, each resulting vector quantization (VQ) index is also conditionally entropy encoded based on a causal set of spatially and spectrally adjacent indices.

Following a different approach, a number of methods that apply a decorrelating transform were developed. In [3], a 3D version of the quadtree-based codec SPECK [2] was introduced. 3D-SPECK takes small portions of the hyperspectral block, e.g., 16 spectral bands at a time, applies a 3D discrete wavelet transform (DWT) and extends the concept of partitioning sets and rules to the three dimensional case. Given the energy compaction properties of the DWT, and SPECK's efficiency in the coding of significance information, the method achieves very good compression results.

In [12] we presented the LVQ-SPECK codec, tailored to deal with multidimensional data. That algorithm considers each spectral vector as a "multidimensional pixel", employing the same encoding steps as defined for SPECK [2], the only changes being the definition of vector significance against a threshold, the existence of a lattice-based vector codebook and a threshold scaling factor α. Basic characteristics of the scalar method such as the embeddedness of the bit stream and quality/rate scalability were retained, being considered as desirable features of the codec.

This article presents two extensions of LVQ-SPECK, namely the ability to encode a number of spectral bands which is a multiple of the dimension of the lattice being used as the basis for the codebook and the use of an additional decorrelating transform, in this case a discrete wavelet transform, applied in the spectral direction, therefore generating a wavelet packet decomposition [11].

This article is organized as follows. Section II presents the basics of successive approximation methods based on lattice vector quantizers. The LVQ-SPECK encoding algorithm [12] is reviewed in section III. Section IV details the necessary modifications in the algorithm to make it work with a larger number of spectral bands, while the results obtained in the compression of standard AVIRIS hyperspectral images appears in section V. Lastly, section VI presents our conclusions and directions for future developments.

II. Successive-approximation Coding of Vectors

This section presents the basic concepts associated with the encoding of vector-valued samples by successive approximation. It follows the description presented in [12].

A structured method for successive refinement of vectors was presented by Mukherjee and Mitra [7], [8], in which scaled versions of a given lattice are used as quantizers over each step of the approximation process. Voronoi region encoding is the basic operation in this framework, and it is performed according to the following concepts:

- Base lattice (Λ_1): lattice coset from which the codebook is actually derived
- Shape lattice (Λ_0): higher scale lattice which determines the shape of the codebook

The resulting quantizer, called *Voronoi Lattice Vector Quantizer*, is therefore defined as

$$\text{VLVQ}(\Lambda_0, \Lambda_1) = V_0(\Lambda_0) \cap \Lambda_1 \quad (1)$$

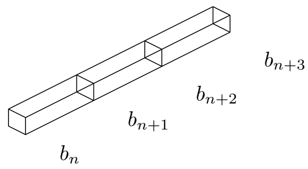


Figure 1. 4-dimensional vector sample $\mathbf{v}(x, y)$

where $V_0(\Lambda_0)$ is the zero-centered Voronoi region associated with the lattice. The shape lattice is defined so that it covers the n-dimensional region of support of the data source and, in the most common case, the base lattice is just a scaled down and (possibly) translated version of the shape lattice, i.e.,

$$\Lambda_1 = \frac{\Lambda_0}{r} - \mathbf{t}, \tag{2}$$

$\mathbf{t}$ being the translation vector.

In a separate development, da Silva and Craizer [5] showed that successive approximation of vectors, under certain conditions, is guaranteed to converge in a finite amount of time. In [5] a vector $\mathbf{v}$ is said to be successively approximated by a sequence of codewords $\mathbf{u}_l$ if the summation

$$\mathbf{v} = \sum_{l=0}^{\infty} \alpha^l \mathbf{u}_l, \tag{3}$$

$$\mathbf{u}_l \in C = \{\mathbf{c}_0, \mathbf{c}_1, ..., \mathbf{c}_K\}$$

converges, where C is the codebook, $\mathbf{c}_k$ are the codewords and α is a scaling factor to account for the fact that after each interaction the residual error is bound by a smaller N-dimensional hypersphere. Every codebook has a choice of α that proves to be optimal, i.e., provides the best representation results.

In lossy coding the infinite summation is substituted for a finite one (since only a good enough approximation of the original data – with a limited amount of error – is desired) resulting in

$$\mathbf{v}_L = \sum_{l=0}^{L} \alpha^l \mathbf{u}_l. \tag{4}$$

LVQ-SPECK carries out this approximation by choosing, at each encoding pass, the one codeword that best represents the residual error between the original data and its current reconstructed version. Our experiments will consider two different four-dimensional codebooks – the 1^{st} and 2^{nd} shells of the D_4 lattice, with the codewords being properly normalized to unit length. The obtained results shall also confirm the superiority of the D_4 shell-2 codebook for the encoding of multidimensional images.

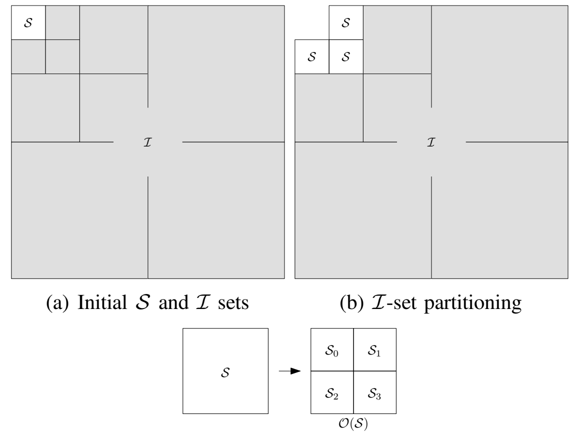
III. LVQ-SPECK

We now present a brief description of LVQ-SPECK [12] consisting of its main procedures, sets definitions and partitioning methods. For more details concerning the method, the reader is referred to [12].

LVQ-SPECK applies a DWT to each of the scalar bands, generating a group of adjacent data sets containing transform coefficients. A *Group Of Images* (GOI) is defined as this set of adjacent transformed spectral bands b_i being encoded. Figure 1 shows how a vector sample $\mathbf{v}(x, y)$ is defined for a given GOI of dimension 4. Hence, for each spatial coordinate, we have

$$\mathbf{v}(x, y) = (b_n(x, y), b_{n+1}(x, y), b_{n+2}(x, y), b_{n+3}(x, y)), \tag{5}$$

where each component belongs to a distinct spectral band. The significance measure is defined as



(a) Initial $\mathcal{S}$ and $\mathcal{I}$ sets (b) $\mathcal{I}$-set partitioning

(c) S-set partitioning

Figure 2. Set definition and partitioning for the SPECK algorithm

$$\Gamma_n(\mathcal{T}) = \begin{cases} 1, & \text{if } \max\limits_{(\text{x,y}) \in \mathcal{T}} \parallel \mathbf{v}(\text{x, y}) \parallel \geq T_n \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

and compares the vector's norm against the current encoding threshold T_n.

LVQ-SPECK uses the same two classes of partitioning sets of SPECK, $\mathcal{S}$ and $\mathcal{I}$ (shown on Figure 2), to convey the significance information of a group of samples. Initially, the $\mathcal{S}$ set is defined to be the set comprised of the low-low frequency subband coefficients of the wavelet transform, with the $\mathcal{I}$ set accounting for all remaining coefficients.

The encoding steps are as follows:

1) **Initialization:**
   - Partition image transform $\mathcal{X}$ into $\mathcal{S}$ and $\mathcal{I} = \mathcal{X} - \mathcal{S}$ sets.
   - The initial threshold T_0 and the threshold scaling factor α are transmitted.
   - Add $\mathcal{S}$ to the LIS, and set LSP $= \emptyset$.
2) **Sorting pass:**
   - for each set $\mathcal{S} \in$ LIS, and in increasing order of size $|\mathcal{S}|$, do `ProcessS`($\mathcal{S}$).
   - if $\mathcal{I} \neq \emptyset$, `ProcessI()`
3) **Refinement pass:**
   - for each (x, y) in the LSP, if the residual norm is larger than the current threshold, output the index of the codeword that best represents it. Otherwise, output the zero-codeword index, since there is no refinement to take place.
4) **Quantization step:**
   - update the encoding threshold, i.e., set $T_n = \alpha \times T_{n-1}$, and go to step 2.

The procedures involved in the encoding/decoding process are defined as follows:

- `ProcessS`($\mathcal{S}$)**:**
  1) output $\Gamma_n(\mathcal{S})$
  2) if $\Gamma_n(\mathcal{S}) = 1$
     - if $\mathcal{S}$ corresponds to a pixel, then output its codebook index and add $\mathcal{S}$ to the LSP
     - else `CodeS`($\mathcal{S}$)
     - if $\mathcal{S} \in$ LIS, then remove $\mathcal{S}$ from LIS
  3) else if $\mathcal{S} \notin$ LIS, then add $\mathcal{S}$ to LIS

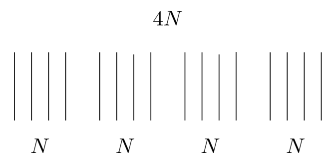


Figure 3. ER LVQ-SPECK Group of Bands to be encoded (each vertical bar represents a spectral band)

4) return

- `CodeS(`$\mathcal{S}$`):`
  1) partition $\mathcal{S}$ into four equal subsets $\mathcal{O}(\mathcal{S})$
  2) for each $\mathcal{S}_i \in \mathcal{O}(\mathcal{S})$
     – output $\Gamma_n(\mathcal{S}_i) = 1$
     – if $\Gamma_n(\mathcal{S}_i) = 1$
       * if $\mathcal{S}_i$ corresponds to a pixel, then output its codebook index and add $\mathcal{S}_i$ to the LSP
       * else `CodeS(`$\mathcal{S}_i$`)`
     – else add $\mathcal{S}_i$ to LIS
  3) return
- `ProcessI():`
  1) output $\Gamma_n(\mathcal{I})$
  2) if $\Gamma_n(\mathcal{I}) = 1$, then `CodeI()`
- `CodeI():`
  1) partition $\mathcal{I}$ into three sets $\mathcal{S}_i$ and one $\mathcal{I}$ (see Fig. 2)
  2) for each $\mathcal{S}_i$, do `ProcessS(`$\mathcal{S}_i$`)`
  3) `ProcessI()`
  4) return

In the upcoming section we shall propose extensions to this method in order to 1) process a larger number of spectral bands simultaneously and 2) incorporate a decorrelating transform in the third dimension direction.

IV. Beyond LVQ-SPECK

A. Extended-Range LVQ-SPECK

In a successive approximation setting such as the one employed by LVQ-SPECK, if large differences of variance exist among the spectral bands, it is expected that the amount of information needed to characterize them will vary greatly as well. In particular, those with small variances usually require smaller rates in order to achieve a given reproduction quality, allowing for some of the remaining bit budget to be used in the refinement of data from larger variance spectral bands. Therefore an overall gain in average performance is expected if, instead of encoding a number of adjacent spectral bands equal to the codeword dimension, we opt to simultaneously encode an integer multiple of that dimension.

Keeping in mind that the codeword dimension is N, and also to simplify the transition into the next proposed extension (the use of a DWT in the spectral direction), we set the number of bands to be encoded as $4N$. We shall denote this version of LVQ-SPECK as Extended-Range LVQ-SPECK (ER LVQ-SPECK).

Once again a 2D DWT is applied to each of the spectral bands separately, resulting in a dataset comprised of $4N$ transformed bands. Each group of N bands will define a *Group of Bands* (GOB), as illustrated in Fig. 3, and each GOB will be processed much the same way as the single GOI is in LVQ-SPECK.

Each GOB will have its own $\mathcal{S}$ and $\mathcal{I}$ sets, with $\mathcal{S}$ being defined as the set containing the LL subbands and $\mathcal{I}$ as its complement in relation to the whole image (recall Figure 2).

Bit allocation among the GOBs is done in a straightforward way. The encoder starts by determining which vector from the full dataset, i.e. the four GOBs, possesses the largest norm. The initial encoding threshold is then calculated from that norm and used to determine vector significance according to the same principles outlined for the LVQ-SPECK. The main difference here resides in the fact that initially there are four different $\mathcal{S}$ and $\mathcal{I}$ sets and that, at any time in the encoding process, the partitioning process in a given GOB does not interfere whatsoever with the remaining ones. It must be stressed, however, that the encoding threshold T_n and the threshold reduction factor α are common to all GOBs.

B. 2D+1D DWT LVQ-SPECK

We propose a second extension of LVQ-SPECK that employs a separate decorrelating transform in the spectral direction. This version, which we shall denote 2D+1D DWT LVQ-SPECK (or 2+1 LVQ-SPECK for short) utilizes a wavelet packet-type decomposition [11] of the extended range dataset presented above prior to the encoding process. The combination of the energy compaction properties of such a decomposition with the use of LVQ-SPECK results in significant performance improvements over all the preceding methods, and illustrates once again the usefulness of such approach for the coding of multidimensional datasets. The steps followed by the encoder can be summarized as:

- Apply spatial 2D DWT to each of the $4N$ subbands
- Apply spectral 1D DWT to each of the $4N$-dimensional vectors that comprise the dataset.
- Scan the dataset (now made up of wavelet packet coefficients) and determine the largest vector norm; establish initial encoding threshold.
- Define initial encoding sets and process them as in ER LVQ-SPECK.

It should be noted that no constraints are placed in the choice of the additional DWT and, in fact, different choices for the spatial and spectral decompositions are often used. Also, while no effort was made here in that regard, the wavelet packet decomposition can be further subjected to an optimization procedure to maximize the coding gain obtained with its use [11].

V. Experimental Results

The ER and 2+1 LVQ-SPECK extensions were used to compress scenes of the AVIRIS hyperspectral images "Moffet Field" and "Jasper Ridge" (obtained from http://aviris.jpl.nasa.gov), both cropped to 512×512×224. A pre-processing step removed all the zero-energy spectral bands, and their indices were sent as overhead. The spectral bands were then grouped into 16-dimensional blocks to be encoded.

The DWT kernel used was the 9/7 wavelet [1], and a 5-stage 2D transform was applied to each spectral band. Simulations involving the 2+1 LVQ-SPECK codec also employed the 9/7 wavelet as the decorrelating transform the spectral bands direction. Bit allocation across subbands is done implicitly based on the significance of each vector being encoded. As in LVQ-SPECK, each significance test accounts for one bit in the final bitstream and, since both 4-dimensional codebooks used contain 24 vectors, in the worst case vector index transmission will demand $\log_2 24 = 4.59$ bits during the sorting pass and $\log_2 25 = 4.64$ bits during the refinement ones (to account for the zero codeword).

Table I presents reconstruction results for the each hyperspectral block, when processed by LVQ-SPECK, ER LVQ-SPECK, 2+1 LVQ-SPECK, the 3D-SPIHT and 3D-SPECK algorithms [4], Multi Com-

Table I
AVERAGE SNR (IN dB) FOR AVIRIS HYPERSPECTRAL IMAGES. (VALUES IN PARENTHESIS INDICATE AVERAGE RMSE)

	Jasper Ridge (scene 01)				Moffet Field (scene 01)				Moffet Field (scene 03)			
Rate(bpppb)	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0	0.1	0.2	0.5	1.0
3D-SPIHT[4]	19.59	23.59	31.48	38.36	16.57	21.46	29.88	38.54	12.92	18.25	27.28	35.62
3D-SPECK[4]	19.7	23.66	31.75	38.55	16.67	21.52	29.91	38.60	12.60	17.98	26.99	35.37
JPEG2000 MC[4]	18.25	22.17	29.81	36.63	15.29	19.92	28.19	36.56	10.79	16.81	25.82	33.44
LVQ-SPECK	15.31	17.13	20.41	24.18	18.46	20.15	23.35	27.18	15.35	17.71	22.57	27.74
(D4-sh1)	(244)	(201.9)	(143.7)	(96.2)	(344.5)	(301.5)	(232.0)	(162.3)	(327.6)	(278.1)	(200.6)	(138.4)
LVQ-SPECK	17.72	20.39	25.65	31.61	20.82	23.39	28.69	35.00	18.72	22.51	29.40	35.16
(D4-sh2)	(189.7)	(144.9)	(84.8)	(46.3)	(283.5)	(232.5)	(156.4)	(97.6)	(255.2)	(201.0)	(135.6)	(91.9)
ER LVQ-SPECK	15.16	16.84	20.20	24.02	17.81	19.45	22.72	26.40	14.50	16.96	21.36	26.45
(D4-sh1)	(245.9)	(205.5)	(141.9)	(92.6)	(361.0)	(302.6)	(202.7)	(125.7)	(340.2)	(274.5)	(171.4)	(95.4)
ER LVQ-SPECK	**17.46**	**20.04**	**25.10**	**31.01**	**20.15**	**22.59**	**27.72**	**33.76**	**17.71**	**21.2**	**27.56**	**33.13**
(D4-sh2)	**(193.6)**	**(147.5)**	**(85.0)**	**(43.4)**	**(294.3)**	**(231.0)**	**(133.4)**	**(66.7)**	**(263.8)**	**(196.1)**	**(110.6)**	**(58.1)**
2+1 LVQ-SPECK	19.00	22.13	27.96	34.22	21.38	24.21	29.99	36.19	19.32	23.14	29.53	34.17
(D4-sh1)	(171.1)	(126.8)	(74.8)	(40.9)	(303.6)	(258.0)	(182.8)	(111.5)	(277.4)	(230.4)	(164.5)	(106.9)
2+1 LVQ-SPECK	**22.12**	**26.09**	**32.39**	**36.76**	**24.56**	**28.49**	**34.58**	**38.56**	**23.13**	**26.98**	**31.41**	**34.49**
(D4-sh2)	**(122.8)**	**(83.8)**	**(44.7)**	**(26.9)**	**(224.9)**	**(174.6)**	**(110.8)**	**(66.0)**	**(213.1)**	**(174.4)**	**(122.2)**	**(77.3)**
2D-SPECK	14.60	16.23	19.28	22.78	17.76	19.47	22.45	25.86	14.61	16.89	21.34	26.43
	(262.0)	(221.1)	(161.1)	(111.8)	(360.8)	(311.0)	(239.2)	(173.8)	(339.3)	(283.8)	(202.6)	(139.2)

ponent feature of JPEG2000[10], and the original 2D-SPECK codec applied to each of the spectral bands individually. The figures of merit utilized here are the average root mean-squared error (RMSE) and the signal-to-quantization noise ratio (SNR), defined as

$$\mathrm{SNR} = 10 \log_{10} \frac{P_x}{\mathrm{MSE}} \mathrm{~dB} \tag{7}$$

where P_x is the power of the original signal and MSE is the reproduction mean-squared error.

Careful observation of the presented RMSE values, which seems to be the most appropriate and significant measure when dealing with hyperspectral images, shows that significant improvements were obtained with the inclusion of both the presented extensions, especially at larger bit rates. In particular, for a rate of 1.0 bpppb and using D_4 shell-2 as the codebook (which consistently presents the best option), average RMSE values for the Moffet3 dataset are reduced from 91.9 in LVQ-SPECK to 58.1 for the ER version and 77.3 for the 2+1 version. Similar trends are observed for Moffet1, whose correspondent values are respectively 97.6, 66.7 and 66.0 and for Jasper1, with 46.3, 43.4 and 26.9, respectively.

VI. CONCLUSIONS

We presented two incremental extensions to the LVQ-SPECK algorithm for encoding hyperspectral images, namely the Extended Range LVQ-SPECK and the 2D+1D DWT LVQ-SPECK, and showed that both options provide a considerable improvement in the performance of the original codec, in terms of reducing the reproduction average RMSE for larger bit rates. The Extended Range LVQ-SPECK attempts to provide better rate allocation by increasing the number of spectral bands being simultaneously encoded, while the 2D+1D DWT LVQ-SPECK additionally exploits the energy compaction properties of a wavelet packet decomposition to improve the overall coding gain.

We expect further improvements with the use of higher-dimensional lattices, such as the E_8, Λ_{16} and Λ_{24} lattices, as the increased dimensionality is bound to provide still better rate allocation throughout the encoding process.

ACKNOWLEDGEMENT

This work was performed at Rensselaer Polytechnic Institute and was partly supported by Fundação CAPES, Brazil, grant No. 1535/98-6. The authors would also like to thank the Brazilian Funding Agency FINEP under contract no. 2645/06 FINEP/FAPEB and the Army Technological Center-Brazil for their financial support.

REFERENCES

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, pp. 205–220, April 1992.

[2] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, pp. 1219–1235, 2004.

[3] X. Tang, W. A. Pearlman, and J. W. Modestino, "Hyperspectral image compression using three-dimensional image coding," in *SPIE/IS&T Electronic Imaging 2003*, vol. 5022 of *Proceedings of the SPIE*, Jan. 2003.

[4] X. Tang and W. A. Pearlman, *Three-Dimensional Wavelet-Based Compression of Hyperspectral Images*, ch. Hyperspectral Data Compression. Kluwer Academic Publishers, 2006.

[5] E. A. B. da Silva and M. Craizer, "Generalized bit-planes for embedded codes," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 2, pp. 317–321 vol.2, 1998.

[6] G. Motta, F. Rizzo, and J. A. Storer, "Compression of Hyperspectral Imagery," in *Proceedings of the Data Compression Conference*, pp. 333–342, March 2003.

[7] D. Mukherjee and S. K. Mitra, "Successive refinement lattice vector quantization," *IEEE Transactions on Image Processing*, vol. 11, pp. 1337–1348, Dec. 2002.

[8] D. Mukherjee and S. K. Mitra, "Vector SPIHT for embedded wavelet video and image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 231–246, Mar. 2003.

[9] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1991.

[10] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image Compression: Fundamentals, Standards and Practice*. (The International Series in Engineering and Computer Science), Kluwer Academic Publishers, 2002.

[11] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 2 ed., 1999.

[12] A. J. S. Dutra, W. A. Pearlman, and E. A. B. da Silva, "Compression of Hyperspectral Images with LVQ-SPECK," in *Proceedings of the Data Compression Conference*, (Snowbird, UT), Mar. 2008.