Multistage Lattice Vector Quantization for Hyperspectral Image Compression

CIPR Technical Report TR-2008-1

Ying Liu and William A. Pearlman

November 2007

Center for
Image Processing Research

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

Multistage Lattice Vector Quantization for Hyperspectral Image Compression

Lattice vector quantization (LVQ) offers substanitial reduction in computational load and design complexity due to the lattice regular structure [2]. In this chapter, we extended SPIHT coding algorithm with lattice vector quantization to code hyperspectral images. In the proposed algorithm, multistage lattice vector quantization (MLVQ) is used to exploit correlations between image slices, while offering successive refinement with low coding complexity and computation. Different LVQs including cubicZ_4 and D_4 are considered. And their performances are compared with other 2D and 3D wavelet-based image compression algorithms.

1 Introduction

The transform, the quantization and the coding of quantized coefficients are all candidates for exploiting the relationships between the slices. Due to the superior performance over scalar quantization, vector quantization has been applied in many wavelet-based coding algorithms.

The Linde-Buzo-Gray (LBG) algorithm [5] is the most common approach to design vector quantizer. In [3], subband image coding with LBG codebook generation VQ is proposed. LBG training algorithm causes high computational cost and coding complexity especially as the vector dimension and bit rate increase. Lattice vector quantization, which is an extension of uniform scalar quantization to multiple dimensions, is an approach to reduce the computational complexity [7].

Plain lattice vector quantization of wavelet coefficient vectors has been successfully employed for image compression [11, 12, 13]. In order to improve performance, it is reasonable to consider combining LVQ with powerful wavelet-based zerotree or set-partitioning image coding methods and bitplane-wise successive refinement methodologies for scalar sources, as in EZW, SPIHT and SPECK. In [14], a multistage lattice vector quantization is used along with both zerotree structure and quadtree structure that produced comparable results to JPEG 2000 at low bit rates. VEZW [15] and VSPIHT [16, 17, 18] have successfully employed LVQ with 2D-EZW and 2D-SPIHT respectively. And in VSPECK [19], tree-structured vector quantization (TSVQ) [20] and ECVQ [21] are used to code the significant coefficients for 2D-SPECK.

For volumetric images, especially for hyperspectral images, neighboring slices convey highly related spatial details. Since VQ has the ability to exploit the statistical correlation between neighboring data in a straightforward manner, we plan to use VQ on volumetric images to explore the correlation in the axial direction. In particular, the multistage LVQ is used to obtain the counterpart of bitplane-wise successive refinement, where successive lattice codebooks in the shape of Voronoi regions of multidimensional lattice are used.

This chapter is organized as follows. We first reviews basic lattice vector quantization and the multistage LVQ. The multistage LVQ-based-SPIHT (MLVQ-SPIHT) is given in Section 3. The performance of the MLVQ-SPIHT for hyperspectral image compression is presented in Section 4. Section 5 concludes the chapter.

2 Vector Quantization

The basic idea of vector quantizer is to quantize pixel sequences rather than single pixels. A vector quantizer of dimension n and size L is defined as a function that maps an arbitrary vector $X \in R^n$ into one of L output vectors $Y_1, Y_2, ..., Y_L$ called codevectors belonging to R^n. The vector quantizer is completely specified by the L codevectors and their corresponding nonoverlapping partitions of R^n called Voronoi regions. A Voronoi region V_i is defined by the equation [22]

$$V_i = \{X \in R^n / \|X - Y_i\| \leq \|X - Y_j\|, i \neq j\} \tag{1}$$

Given a desired bit rate per dimension b and vector dimension n, the codebook size is equal to 2^{bn}. Although the LBG (Lind-Buzo-Gray) algorithm can generate locally optimal codebooks, the complexity grows exponentially because the codewords have to be compared and chosen between 2^{bn} possible vectors.

Lattice Vector Quantization (LVQ), which builds codebooks as subsets of multidimensional lattices, solves the complexity problem of LBG based vector quantizer and yields very general codebooks.

2.1 Lattice Vector Quantization

A lattice L in R^n is composed of all integral combinations of a set of linearly independent vectors. That is

$$L = \{Y | Y = u_1 a_1 + ... + u_n a_n\} \tag{2}$$

where $\{a_1, ..., a_n\}$ is a set of n linearly dependent vectors, and $\{u_1, ..., u_n\}$ are all integers. A lattice coset Λ, is obtained from a lattice L by adding a fixed translation vector t to the points of the lattice

$$\Lambda = \{Y | Y = u_1 a_1 + ... + u_n a_n + t\} \tag{3}$$

Around each point Y_i in a lattice coset Λ, its Voronoi region is given as

$$V(\Lambda, Y_i) = \{X \in R^n / \|X - Y_i\| \leq \|X - Y_j\|, Y_i \in \Lambda, \forall Y_j \in \Lambda\} \tag{4}$$

The zero-centered Voronoi region $V(\Lambda, 0)$ is defined as

$$V(\Lambda, 0) = V(\Lambda, Y_i) - Y_i \tag{5}$$

In lattice vector quantization, the input vector is mapped to the lattice points of a certain chosen lattice type. The lattice points or codewords may be selected from the coset points or the truncated lattice points [22].

2.1.1 Classical Lattice

Conway and Sloane [7] investigated the lattice properties and determined the optimal lattices for several dimensions. They also give a fast quantization algorithm [8] which makes searching the closest lattice point to a given vector extremely fast.

The cubic lattice Z^n is the simplest lattice form which consists all the integer points in the coordinate system with a certain lattice dimension. Other important lattices are the root lattices$A_n(n \geq 1)$, $D_n(n \geq 2)$, $E_n[n = 6, 7, 8]$ and the Barnes-Wall lattice $\wedge_{16}$. These lattices give the best sphere packings and coverings in their respective dimension [2]. In this chapter, we have used theZ^4, D^4 and A^2. The A_n lattice is defined by the following:

$$A_n = \{(x_0, x_1, ..., x_n) \in Z_{n+1} : x_0 + ... + x_n = 0\}, For\ n \geq 1 \tag{6}$$

For $n \geq 3$, the D_n lattice is defined as the follows:

$$D_n = \{(x_1, ..., x_n) \in Z_n : x_1 + ... + x_n\ even\} \tag{7}$$

The lattice E_8 is defined by

$$E_8 = D_8 \bigcup (\frac{1}{2}\mathbf{1} + D_8) \tag{8}$$

where **1** stands for all-one vector.

2.1.2 LVQ Codebook

The codebook of a lattice quantizer is obtained by selecting a finite number of lattice points out of an infinite lattice. An LVQ codebook is decided by a root lattice, a truncation and a scaling factor. The root lattice is the lattice coset from which the codebook is actually constructed. A truncation must be applied on a root lattice in order to select a finite number of lattice points and quantize the input data with finite energy. The bit rate of the LVQ is determined by the number of points in the truncated area. To obtain the best distortion rate, we must scale and truncate the lattice properly. To do this, we need to know how many lattice points lie within the truncated area, i.e. to know the shape of the truncated area.

Two kinds of truncation shapes are consider in this chapter. When the signal to be compressed has an i.i.d. multivariate Gaussian distribution, the surfaces of equal probability are ordinary spheres. The truncated area is spherical [12]. In these applications the size of the codebook was calculated by the theta function of lattice, which was described in [2]. In the case of Laplacian sources (for cubic lattice) where surfaces of equal probability are spheres for theL^1 metric, which are sometimes called pyramids. The number of lattice points $Num_(n, r)$ lying on a hyper-pyramid of radius r in n-dimensional space R^n is given by Fischer [10] as:

$$Num(n, r) = Num(n - 1, r) + Num(n - 1, r - 1) + Num(n, r - 1) \tag{9}$$

The truncation is determined by specifying the shape and radius of the hypersphere/hyperpyramid that best matches the probability distribution of the input source. The scaling factor is used to control the distance between any two nearest lattice points, i.e. the maximum granular error of the quantizer [14]. The support of the distribution of the granular quantization error has the shape of the Voronoi region.

2.2 Multistage Lattice Vector Quantization

The essence of our successive refinement lattice VQ is to generate a series of decreasing scale zero-centered Voronoi lattice regions, $V_0(\Lambda_0, 0), V_1(\Lambda_1, 0), V_2(\Lambda_2, 0), ...$, each covering the zero-centered Voronoi region of the previous higher scale. The highest scale quantizer is completely specified by lattice points y_i and its corresponding nonoverlapping Voronoi region $V_0(\Lambda_0, y_i)$. To prevent divergence of overload quantization error, the truncated LVQ at current stage should be able to cover the Voronoi region of the previous stage. On the other hand, any overlap of quantization regions at two successive stages will decrease compression efficiency. So the optimal truncated lattice should be consistent with the Voronoi region of the root lattice [14]. However, this optimal condition can not be always satisfied.

Figure 1 gives an example of this multistage LVQ with the hexagonal A_2 lattice and scale-down factor $r = 4$. First, input vector $x \in R_n$ is quantized to be output vector $u_0 = y_2$ by the highest scale quantizer. The uncertainty in x has been reduced to the Voronoi region $V_0(\Lambda_0, y_2)$ around the chosen codevector y_2. Next quantizer quantizes the approximation error $(x - u_0)$, which falls into the zero-centered Voronoi region $V_0(\Lambda_0, y_0)$, using lower lattice VQ quantizer to obtain a refinement $u_1 = z_i$. Now, the uncertainty in x is reduced to the zero-centered Voronoi region $V_1(\Lambda_1, 0)$ of lattice coset V_1. The next lower scale quantizer quantizes the error $(x - u_0 - u_1)$ reducing the uncertainty in x to the zero-centered Voronoi region of V_2. Continuing in this way, the final approximation $\hat{x}$ of vector x is

$$\hat{x} = u_0 + u_1 + u_2 + ...$$

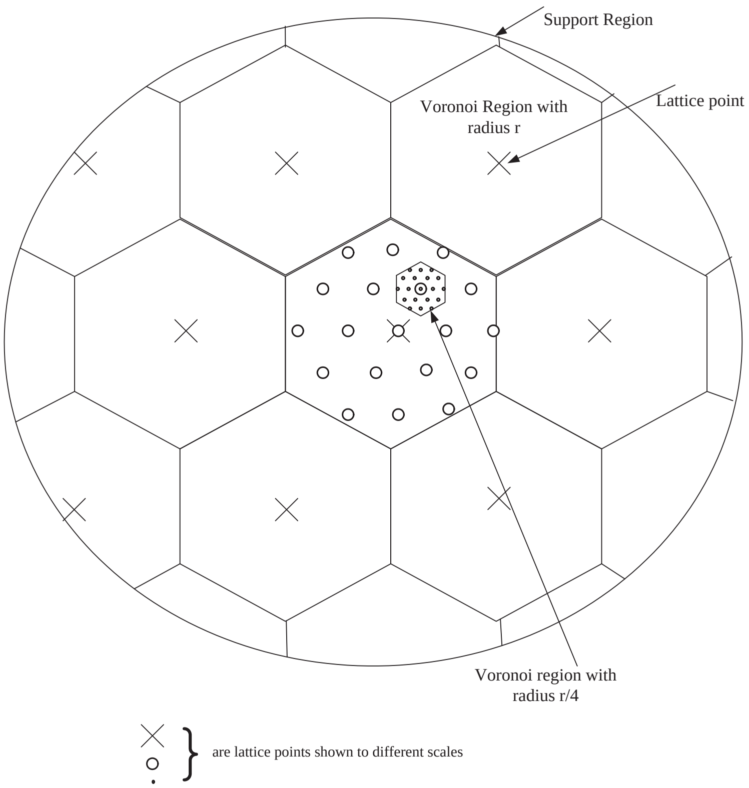
Support Region
Lattice point
Voronoi Region with radius r
Voronoi region with radius r/4
are lattice points shown to different scales

Figure 1: Multistage lattice VQ with A_2 lattice.

3 MLVQ-SPIHT

In this section, we describe our new algorithm MLVQ-SPIHT, which combines multistage lattice vector quantization methodology and the SPIHT coding algorithm to code 3D hyperspectral image data sets.

In MLVQ-SPIHT, 2D DWT is applied on each image slice independently. For a given vector dimension N, we segment the image sequence of the transformed images into groups, GOS $=N$. $N = 4$ is used for our application. The coding algorithm will be applied on every GOS independently. For every transformed image slice in the same GOS, we group wavelet coefficients at the same spatial location into vectors. For example, for spatial location (i, j) and transformed slices $S_1, S_2, ..., S_N$ in the GOS, the vector associated with this location is $v(i, j) = (S_0(i, j), S_1(i, j), ..., S_{N-1}(i, j))$. A parent-child relationship between the vectors in different subbands is the same as in [1]. Figure 2 gives an example of parent-child relationship between vectors when vector dimension$N = 4$.

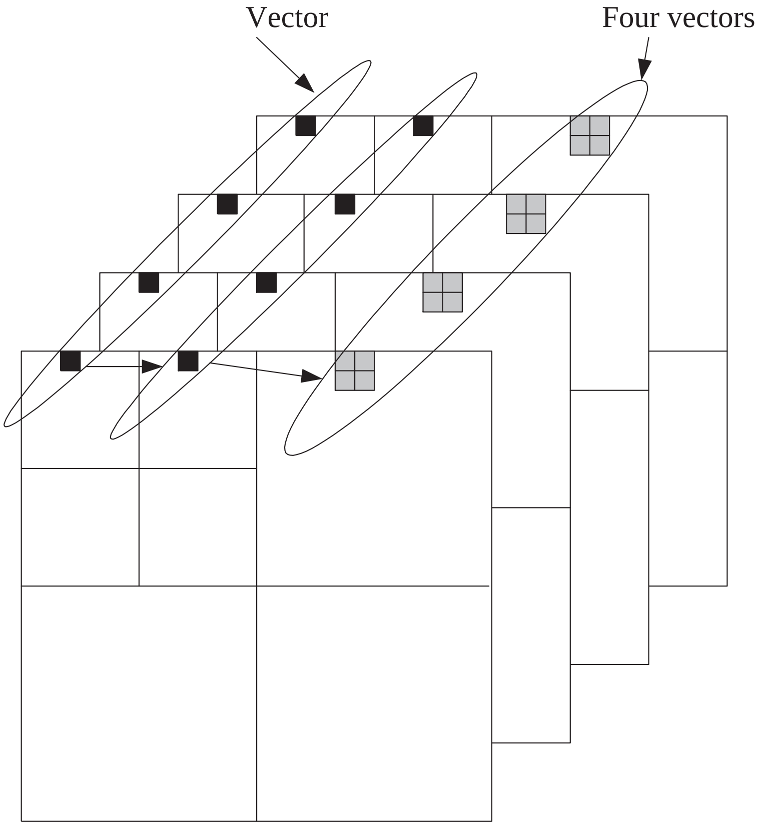

Figure 2: An example of parent-child relationship between vectors when vector dimension$N = 4$.

The SPIHT algorithm is used to search for significance at the current metric threshold, which is based on certain pre-defined decision regions that gradually decrease in scale following a given rule. Every decision region is defined by two surfaces (one is on the inside, the other is one the outside) enclosing the origin that successively decrease in size. For every given decision region, the SPIHT algorithm is used to test the significance of the N-dimensional vectors. Each sorting pass locates significant vectors and roughly quantizes those significant vectors in the same pass. The vectors ascertained as significant in a pass will be progressively refined in successive passes using our multistage LVQ. Figure 3 uses the

A_2 lattice to illustrate our vector SPIHT, where the lattice at each stage decreases in scale by a factor $r = 4$, the threshold for SPIHT sorting pass decreases in scale by a factor of 2, and theL_2 norm is used for significant test. The wavelet vectors are first scaled so that all scaledL_2 norm vectors will lie within or on the hyperspherical surface of L_2 norm equal to a given standardized value R. For the first sorting pass, the significant region is bounded on the outside by the hyperspherical surface ofL_2 norm R, and on the inside by the hyperspherical surface ofL_2 norm $R/2$. For the following sorting passes, the significant regions are bounded by zero-centered hyperspherical regions, with the inside one having half scale of the outside one. For example, if a vector is ascertained as significant in the first sorting pass, i.e., that vector is located in the 1st significant region, the vector is roughly encoded by the first stage LVQ of the 1st significant region, which uses translations of$V/2$ lattice coset. When the sorting pass reaches the 3rd significant region, the vector ascertained as significant in the 1st pass will be refined by the second stage LVQ, which uses translations of$V/8$ lattice coset. As shown in Figure 3, the bracketed sequences denote the successively lower scale lattices used to quantize vectors in that significant region. We believe this scheme can provide good compression performance with successive refinement.

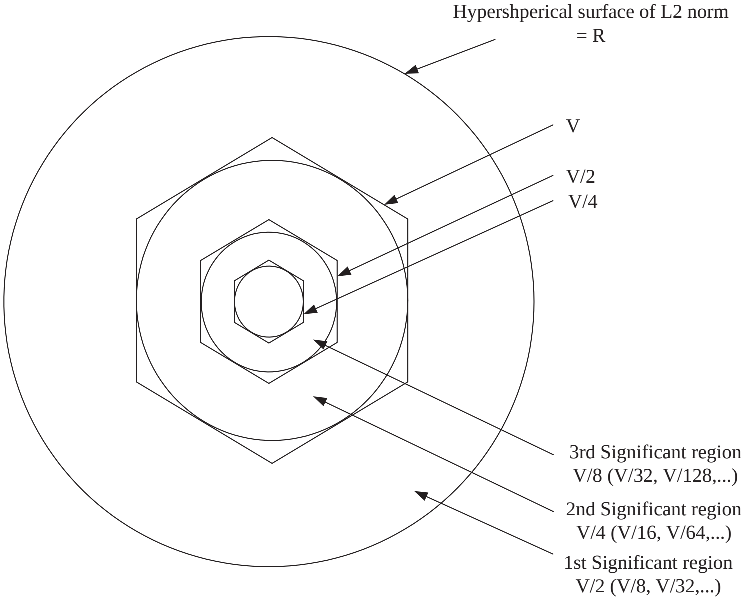

Figure 3: Vector SPIHT with successive refinement LVQ.

Based on the above scheme, we implemented a SPIHT-based coding algorithm using four-dimensional wavelet vectors as shown in Figure 2. Several LVQs are implemented in our scheme.

3.0.1 Cubic Z_4 LVQ

To define a cubic Z_4 LVQ codebook, the root lattice Z_4 and a cubic truncation are used. The cubic truncation requires the L_∞ norm (maximum norm) for vector magnitude measurement. TheL_∞ norm is defined as

$$\|X\|_\infty := max(|x_1|, ...|x_N|)$$

For cubic truncation, the bit rate is evenly allocated to each of the lattice's N-dimension components. That implies Cubic Z_4 LVQ is actually equivalent to four individual scalar quantizers that are applied independently to each of the four coefficient in a vector. The cubic truncation area has exactly the same shape as the Voronoi region of the corresponding root lattice. And the number of codewords in the codebook can always be an integer power of 2, which prevents loss in coding efficiency. Cubic truncation does not well match the typical distributions of subband coefficients and decreases the compression performance. Two different bit rates are used in our cubic Z_4 multistage LVQ. When a newly significant vector is quantized in its first stage quantizer, an 8-bit LVQ is used to quantize both significance and signs. In its all refinement stages, a 4-bit LVQ is used. So the truncation radius of these two kinds of LVQs are 2 and 1, respectively. If the threshold is scaled down by two at each successive layer, these layers are equivalent to bit planes.

3.0.2 Pyramid D_4 LVQ

To define a pyramid D_4 LVQ codebook, the root lattice D_4 and a pyramid truncation are used. The pyramid truncation requires the L_1 norm for vector magnitude measurement. The L_1 norm is defined as

$$\|X\|_1 := \sum_{i=1}^{N} |x_i|$$

In our implementation, the truncation radius is set to four. Lattice points inside this truncation area lie on two hyper-pyramid surfaces with constant L_1 norm 2 and 4, respectively. The number of lattice points on these two shells are 32 and 192, respectively [12]. So 8-bit indexes are used to code these 225 codewords. The same LVQ codebook is used in all stages. Since the Voronoi region is determined by mean square error as shown in Equation 4, and each Voronoi region of D_4 has 24 faces, the Voronoi region is closer to a sphere and is inconsistent with the shape of the pyramid truncation. Therefore, to get the best balance between overlaps and gaps between the Voronoi region at the current stage and that of the previous stage, the scale-down factor is set to 1/3 [14].

3.0.3 Sphere D_4 LVQ

To define a sphere D_4 LVQ codebook, the root lattice D_4 and a sphere truncation are used. The sphere truncation requires the L_2 norm for vector magnitude measurement. The L_2 norm is defined as

$$\|X\|_2 := \sqrt{\sum_{i=1}^{N} |x_i|^2}$$

In our implementation, the truncation radius is set to 2. Lattice points inside this truncation area lie on two hyper-sphere surfaces with constant L_2 norm $\sqrt{2}$ and 2, respectively. The number of lattice points on these two shells are 24 and 24, respectively [12]. So 6-bit indexes are used to code these 48 codewords. The scale-down factor is set to 1/2.

4 Experimental Results

The proposed MLVQ-SPIHT algorithm is used compress hyperspectral image ”Moffet Field”. It’s property is shown in Table 1.

File Name	Image Type	Volume Size	Bit Depth (bit/pixel)	Power (σ_x^2)
moffett scence 3	AVIRIS	$512 \times 512 \times 224$	16	2177316

Table 1: Description of the image volume Moffett Field

The pyramid wavelet decomposition employed here uses the S+P wavelet filter, and a 5-level spatial transform is performed. After wavelet transformation, the magnitude of each vector is calculated according to the norm corresponding to the particular LVQ. The fast quantizing and coding algorithm proposed by Conway and Sloane [8, 9] is used to code the significant vectors. For each significant region, the LVQ indices are coded using an adaptive arithmetic coder and the significant information is adaptively arithmetic coded as described in [1].

Table 2 compares the rate-distortion results for MLVQ-SPIHT using different LVQs with 3D-SPIHT, 3D-SPECK and JPEG2000 multi-component integer implementation for the Moffett hyperspectral image volume [23]. Five-levels of the dyadic S+P (B) integer filter were applied on all three dimensions for 3D-SPIHT and 3D-SPECK. For JPEG2000 multi-component, five-level 1D S+P (B) filter was first applied on spectral axis followed by (5,3) filter on spatial domain. The results show that at low bit rate, MLVQ-SPIHT algorithms outperforms 3D compression algorithms. As the bit rate increases, 3D algorithms give better performance. And in general, sphereD_4 LVQ and pyramid D_4 LVQ show better performance than cubic Z_4 LVQ.

Bit Rate	3D-SPIHT	3D-SPECK	JP2K-Multi	MLVQ-SPIHT Cubic Z_4	MLVQ-SPIHT Pyramid D_4	MLVQ-SPIHT Sphere D_4
0.1 bpp	10.828	10.622	10.264	11.817	11.807	12.361
0.2 bpp	16.740	16.557	15.952	17.144	18.149	18.278
0.5 bpp	26.102	25.998	25.298	23.859	24.605	25.100
1.0 bpp	34.946	34.845	33.835	31.169	31.152	31.305

Table 2: Comparison of rate-distortion results of different coding methods in Signal-to-Noise ration (SNR) in dB

5 Summary and Conclusions

In this chapter, we presented a multidimensional image compression algorithm which is an extension of SPIHT with lattice vector quantization and support successive refinement. In the proposed algorithm, multistage lattice vector quantization is used to exploit correlations between image slices. CubicZ_4 LVQ, sphere D_4 LVQ and pyramid D_4 LVQ are implemented in the proposed scheme. The experimental results show that MLVQ-based-schemes provide better rate-distortion performance at low bit rate than those algorithms that employ 3D wavelet transforms.

6 REFERENCES

References

[1] A. Said and W.A. Pearlman, “A new, fast and efficient image codec based on set-partitioning in hierarchical trees”, *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 6, pp. 243-250, June 1996.

[2] J.H. Conway and N.J.A Sloane, *Sphere-Packing, Lattice, and Groups*, Springer, New York, NY, USA, 1988.

[3] S.P. Voukelatos and J. Soraghan, ”Very low bit-rate color video coding using adaptive subband vector quantization with dynamic bit allocation”, *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 7, No. 2, April 1997, pp. 424-428.

[4] C.E. Shannon, ”Coding theorems for a discrete source with a fidelity criterion” *IRE Nat. Conv. Rec., pt. 4*, 1959, pp. 142-163.

[5] Y.L. Linde, A. Buzo and R.M. Gray, ”An algorithm for vector quantizer design”,*IEEE Trans. on Communication*, Vol. COM-28, Jan. 1980, pp. 84-95.

[6] R.M. Gray, *Source Coding Theoty*, Boston: Kluwer, 1990.

[7] J.H. Conway and N.J.A. Sloane, ”Voronoi region of lattices, second moments of polytopes, and quantization”, *IEEE Trans. on Information Theory*, Vol. IT-28, Mar. 1982, pp. 211-226.

[8] J.H. Conway and N.J.A Sloane, ”Fast quantizing and decoding algorithms for lattice quantizers and codes”, *IEEE Trans. on Information Theory*, Vol. IT-28, Mar. 1982, pp.227-232.

[9] J.H. Conway and N.J.A Sloane, ”A fast encoding method for lattice codes and quantizers”,*IEEE Trans. on Information Theory*, Vol. IT-29, No. 6, Nov. 1983, pp.820-824.

[10] T.R. Fischer, ”A pyramid vector quantizer”, *IEEE Trans. on Information Theory*, Vol. IT-32, July 1986, pp. 568-583.

[11] M. Antonini, M. Barlaud, and P. Mathieu, ”Image coding using lattice vector quantization of wavelet coefficients”, *in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Toronto, ON, Canada, May 1991, pp.2273-2276.

[12] M. Barlaud, P. Sole, T. Gaidon, M. Antonini and P. Mathieu, ”Pyramidal lattice vector quantization for multiscale image coding”, *IEEE Trans. on Image Processing*, Vol. IP-3, No.4, July 1994, pp. 367-381.

[13] A. Woolf and G. Rogers, ”Lattice vector quantization of image wavelet coefficient vectors using a simplified form of entropy coding”, *in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing* Vol. 5, Adelaide, Australia, Apr. 1994, pp. 269-272.

[14] H. Man, F. Kossentini, and M. J. T. Smith, ”A family of efficient and channel error resilient wavelet/subband image coders”, *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, no. 1, pp. 95-108, 1999.

[15] E.A.B. Da Silva, D.G.Sampson, and M. Ghanbari, ” A successvie approximation vector quantizer for wavelet transform image coding”, *IEEE Trans. Image Processing*, vol. 5, pp. 299-310, Feb. 1996.

[16] J. Knipe, X. Li,and B. Han, ”An improved lattice vector quantization scheme for wavelet compression”, *IEEE Trans. Signal Processing*, vol. 46, pp. 239-243, Jan. 1998.

[17] D. Mukherjee and S. K. Mitra, ” Vector set partitioning with classified successive refinement VQ for embedded wavelet image and video coding”, *in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 5, Seattle, WA, May 1998, pp. 2809-2812.

[18] D. Mukherjee and S. K. Mitra, ” Successive refinment lattice vector quantization”, *IEEE Trans. Image Porcessing*, vol. 11, no. 12, pp. 1337-1348, Dec. 2002.

[19] C. C. Chao and R. M. Gray, ”Image compression with vector SPECK algorithm”, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'06)*, vol. 2, pp. 445-448, Toulouse, France, May 2006.

[20] K. Rose, D. Miller, and A. Gersho, ” Entropy-constrained tree-structured vector quantizer design”, *IEEE Trans. Image Porcessing*, vol. 5, No. 2, pp. 393-398, Feb 1996.

[21] P. A. Chou, T. Lookabaugh, and R. M. Gray, ” Entropy-constrained vector quantization”, *IEEE Trans. Acoust. Speech, Signal Processing*, vol. 37, No. 1, , pp. 31-42, Feb 1989.

[22] A.A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic, New York, NY, USA, 1992.

[23] X. Tang and W. A. Pearlman, ”Three-Dimensional Wavelet-Based Compression of Hyperspectral Images”, Chapter in Hyperspectral Data Compression, Kluwer Academic Publishers 2005.