PARTITIONED VECTOR QUANTIZATION: APPLICATION TO LOSSLESS COMPRESSION OF HYPERSPECTRAL IMAGES

Giovanni Motta, Francesco Rizzo and James A. Storer

Computer Science Department, Brandeis University, Waltham MA 02454
{gim, frariz, storer}@cs.brandeis.edu

ABSTRACT

A novel design for a vector quantizer that uses multiple codebooks of variable dimensionality is proposed. High dimensional source vectors are first partitioned into two or more subvectors of (possibly) different length and then, each subvector is individually encoded with an appropriate codebook. Further redundancy is exploited by conditional entropy coding of the subvectors indices. This scheme allows practical quantization of high dimensional vectors in which each vector component is allowed to have different alphabet and distribution. This is typically the case of the pixels representing a hyperspectral image. We present experimental results in the lossless and near–lossless encoding of such images. The method can be easily adapted to lossy coding.

1. INTRODUCTION

In the last two decades, space borne and air borne remote acquisition of high definition images has been increasingly used in military and civilian applications to recognize objects and classify materials on the earth surface. In hyperspectral photography pixels cover a specified area and record the reflected light spectrum decomposed into many bands. For example, images acquired with the NASA *Airborne Visible/Infrared Imaging Spectrometer* (AVIRIS [1, 5]), have each pixel covering an area of approximately 20m x 20m and the reflected light is decomposed into 224 bands. Since every material reflects sun light in its own peculiar way, the analysis of its reflected light spectrum can be used to recognize it. The acquisition of hyperspectral images produces large amounts of highly correlated data in the form of a two dimensional image matrix in which every pixel consists of a vector having hundreds of components, one for each spectral band. Increasing the number of bands, i.e. the spectral resolution, allows for more sophisticated analyses. Augmenting the spectral resolution is advantageous with respect, for example, to the increase in spatial resolution since it only increases the data rate by a linear amount.

Traditional approaches to the compression of hyperspectral imagery are based on differential prediction via DPCM [2] or on dimensionality reduction with methods that use the Principal Component Analysis. PCA is used to compact the representation by isolating the smallest number of independent components in a vector. In practice however, DPCM–based methods are too simple to exploit properly the multidimensional nature of the data and PCA–based lossless compressors do not achieve very good results because the transformed signal is harder to entropy code. The use of PCA in lossy algorithms is also restricted to the use of squared error as distortion measure. Since hyperspectral imagery is acquired at great cost and mostly used in critical tasks like classification (assignment of a label to every pixel) or target detection (identification of a somewhat rare instance) the use of squared error can be sometimes too restrictive.

2. SOURCE CHARACTERIZATION

In the following, we will assume that a hyperspectral image is a discrete time, discrete values bi dimensional random source $\mathbb{I}(x, y)$ that emits D–dimensional vectors $\mathbf{I}(x, y)$. Each vector component $I_i(x, y)$, $0 \le i \le D-1$ is drawn from the alphabet χ_i and is distributed according to a space variant probability distribution that may depend on the other components. The vector quantizer that we wish to design operates on one vector at the time so, in order to simplify the problem, we assume at first that source realizations are independent. The quantizer only removes correlation existing among components of the same vector. The spatial correlation is later exploited with the use of an entropy encoder.

3. PARTITIONED VECTOR QUANTIZER

The complexity of building a vector quantizer (VQ) codebook is known to be computationally prohibitive [3] when the vector dimensionality is large, as it is in hyperspectral imagery. A solution to this problem is the design of a *partitioned* VQ, i.e. the design of a VQ in which input vectors are partitioned into a number of consecutive segments (blocks or subvectors) that are independently quantized. While this leads to a sub–optimal solution in terms of Mean Squared Error (or MSE) because the scheme does not exploit correlation among subvectors, the resulting design turns out to be practical and coding and decoding present a number of advantages in terms of speed, memory requirements and exploitable parallelism.

In our method we divide input vectors into N subvectors and quantize each of them with an L–levels exhaustive search VQ. Since the components of $\mathbf{I}(x, y)$ are drawn from different alphabets, their distributions may be significantly different and partitioning the D components into N blocks of (approximately) equal size may not be optimal. We wish to determine the size of the N sub vectors adaptively, while minimizing the quantization error, measured for example in terms of MSE. Once the N codebooks are designed, input vectors are encoded by partitioning them into N subvectors of appropriate length, each of which is quantized independently with the corresponding VQ. The index of the partitioned codevector is given by the concatenation of the indices of the N subvectors (Figure 1).

Formally, the Partitioned Vector Quantizer (or PVQ) is an N–tuple $\mathbb{Q} = (Q_1, Q_2, \ldots, Q_N)$ of N, L–levels, d_i–dimensional Exhaustive Search Vector Quantizers $Q_i = (A_i, F_i, P_i)$, such that $\sum_{1 \le i \le N} d_i = D$ and:

- $A_i = \{\mathbf{c}_1^i, \mathbf{c}_2^i, ..., \mathbf{c}_L^i\}$ is a finite indexed subset of $\mathbb{R}^{d_i}$ called codebook. Its elements $\mathbf{c}_j^i$ are the code vectors;

- $P_i = \{S_1^i, S_2^i, ..., S_L^i\}$ is a partition of $\mathbb{R}^{d_i}$ and its equivalence classes (or *cells*) S_j^i satisfy:

$$\bigcup_{j=1}^{L} S_j^i = \mathbb{R}^{d_i} \text{ and } S_h^i \cap S_k^i = \varnothing \text{ for } h \neq k \, ;$$

- $F_i : \mathbb{R}^{d_i} \rightarrow A_i$ is a function defining the relation between the codebook and the partition as
$$F_i(\mathbf{x}) = \mathbf{c}_j^i \text{ if and only if } \mathbf{x} \in S_j^i \, .$$
And the index j of the centroid $\mathbf{c}_j^i$ is the result of the quantization of the d_i–dimensional subvector $\mathbf{x}$, i.e. the information that is sent to the decoder.

The design of this vector quantizer aims at the joint determination of the $N+1$ partition boundaries $b_0 = 0 \le b_1 \le \ldots \le b_N = D$ and to the design of the N vector quantizers having dimension $d_i = b_i - b_{i-1}$, $1 \le i \le N$.

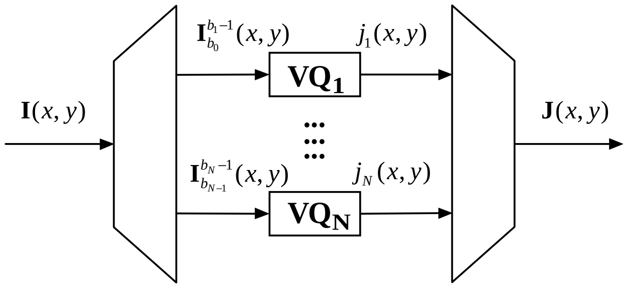

Figure 1: Partitioned Vector Quantizer.

Given a source vector $\mathbf{I}(x, y)$, we indicate the i–th subvector of boundaries b_{i-1} and $b_i - 1$ with the symbol $\mathbf{I}_{b_{i-1}}^{b_i-1}$ (for simplicity, the x and y spatial coordinates are omitted when clear from the context). The mean squared quantization error between the vector $\mathbf{I}$ and its quantized representation $\hat{\mathbf{I}}$, is given by:

$$\begin{aligned} \left(\mathbf{I} - \hat{\mathbf{I}}\right)^2 &= \sum_{i=1}^{N} \left(\mathbf{I}_{b_{i-1}}^{b_i-1} - \hat{\mathbf{I}}_{b_{i-1}}^{b_i-1}\right)^2 \\ &= \sum_{i=1}^{N} \left(\mathbf{I}_{b_{i-1}}^{b_i-1} - \mathbf{c}_{j_i}^i\right)^2 \\ &= \sum_{i=1}^{N} \sum_{h=b_{i-1}}^{b_i-1} \left(I_h - c_{j_i, h-b_{i-1}}^i\right)^2 \end{aligned}$$

Where $\mathbf{c}_{j_i}^i = \left(c_{j_i,0}^i, \ldots, c_{j_i,d_i-1}^i\right)$ is the centroid of the i–th codebook that minimizes the reconstruction error on $\mathbf{I}_{b_{i-1}}^{b_i-1}$:

$$j_i = \arg\min_{1 \le l \le L} MSE\left(\mathbf{I}_{b_{i-1}}^{b_i-1}, c_l^i\right).$$

4. PVQ DESIGN

The design of a PVQ is derived from a variation of the Generalized Lloyd Algorithm (or GLA) described in [4]. Unconstrained vector quantization can be seen as the joint optimization of an encoder (the function $F : \mathbb{R}^d \rightarrow A$ described before) and a decoder (the determination of the centroids for the equivalence classes of the partition $P = \{S_1, S_2, ..., S_L\}$). GLA is an iterative algorithm that, starting from the source sample vectors chooses a set of centroids and optimizes in turns encoder and decoder until the improvements on a predefined distortion measure are negligible. To define our PVQ, the boundaries of the vector partition $b_0 = 0 \le b_1 \le \ldots \le b_N = D$ need to be determined as well. The proposed design follows the same spirit of the GLA. The key observation is that once the partition boundaries are kept fixed, the MSE is minimized independently for each partition by applying the well–known optimality conditions on the centroids and on the cells.

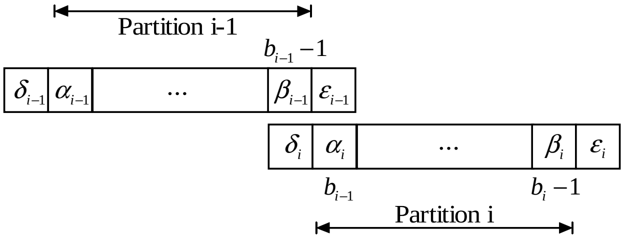

Figure 2: Error contributions for two adjacent partitions.

```
M_i = min( β_{i-1}+α_i , β_{i-1}+ε_{i-1} , δ_i+α_i );
if ( M_i = δ_i+α_i )
        b_{i-1} = b_{i-1} - 1;
else if ( M_i = β_{i-1}+ε_{i-1} )
        b_{i-1} = b_{i-1} + 1;
```

Figure 3: Partition changes in modified GLA.

Vice versa, when the centroids and the cells are held fixed, the (locally optimal) partitions boundaries can be determined in a greedy fashion. The GLA step is applied on each partition independently. The determination of the equivalence classes is performed as usual, but when computing the new centroids, the error for two extra components (one to the left and the other to the right of the boundaries of the current partition) is computed as well. The only exceptions are obviously the leftmost and the rightmost partitions for which only one extra component will be available.

In the iteration involving the i–th partition ($2 \le i \le N-1$), we indicate with

$$\alpha_i = \sum_{x,y}\left(I_{b_{i-1}}(x,y) - \hat{I}_{b_{i-1}}(x,y)\right)^2$$

$$\beta_i = \sum_{x,y}\left(I_{b_i-1}(x,y) - \hat{I}_{b_i-1}(x,y)\right)^2$$

the contribution to the quantization error of its leftmost and rightmost components (see also Figure 2). The two extra components mentioned before are indicated by:

$$\delta_i = \sum_{x,y}\left(I_{b_{i-1}-1}(x,y) - \hat{I}_{b_{i-1}-1}(x,y)\right)^2$$

$$\varepsilon_i = \sum_{x,y}\left(I_{b_i}(x,y) - \hat{I}_{b_i}(x,y)\right)^2$$

The reconstruction values used in the expressions for δ_i and ε_i are determined by the classification performed on the components $b_{i-1},\ldots,b_i$.

The boundary b_{i-1} between the partitions $i-1$ and i is changed according to the criteria expressed by the pseudocode in Figure 3.

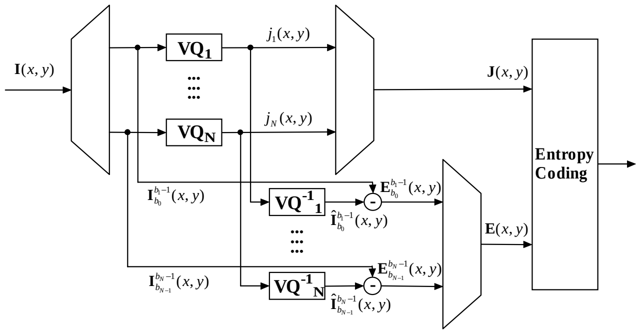

Figure 4: PVQ–based lossless encoder.

5. LOSSLESS CODING

PVQ can be used to perform lossless and near–lossless coding of the source vectors. In this configuration, the quantization is used as a tool to implement dimensionality reduction on the source. Partitioned VQ is combined with inter and intra subvector entropy coding. The quantization residual is entropy coded conditioned on the subvector indices. Figure 4 depicts the encoder that we have used in our experiments.

The partition boundaries and the N unconstrained VQs are determined off–line by the design described earlier. The encoding consists in partitioning the input vector, quantizing the subvectors in order to determine the vector of indices $\mathbf{J}(x,y)$, then computing the quantization error:

$$\mathbf{E}(x,y) = \left(\mathbf{E}_{b_0}^{b_1-1}(x,y), \mathbf{E}_{b_1}^{b_2-1}(x,y), \ldots, \mathbf{E}_{b_{N-1}}^{b_N-1}(x,y)\right)$$

where, for each $1 \le i \le N$:

$$\mathbf{E}_{b_{i-1}}^{b_i-1}(x,y) = \mathbf{I}_{b_{i-1}}^{b_i-1}(x,y) - \hat{\mathbf{I}}_{b_{i-1}}^{b_i-1}(x,y).$$

Since the unconstrained quantizers work independently from each other and independently on each source vector, an entropy encoder is used to exploit this residual redundancy. In particular, each VQ index $j_i(x,y)$ is encoded conditioning its probability with respect to a set of causal indices spatially and spectrally adjacent. The components of the residual vector $\mathbf{E}_{b_{i-1}}^{b_i-1}(x,y)$ are also entropy coded with their probability conditioned on the VQ index $j_i(x,y)$.

6. EXPERIMENTS

The partitioned VQ has been tested on a set of five AVIRIS images downloaded from the NASA web site [5]. AVIRIS images are 614 pixels wide and several thousand pixels high, obtained by flying the spectrometer over the target area. Each pixel represents the light reflected by a 20m x 20m area (high altitude) or 4m x 4m area (low altitude). The spectral response of the reflected light is decomposed into 224 contiguous channels, approximately

10nm wide and spanning from visible to near infrared light (400 to 2500nm). Spectral components are represented with a 16 bits precision.

Several experiments have been performed for various numbers of partitions and for different codebook sizes. The results that we describe here were obtained for $N = 16$ partitions and $L = 256$ codebook levels. The choice of the number of levels makes also practical the use of off–the–shelf image compression tools that are fine–tuned for 8 bit data.

7. RESULTS

Table I shows PVQ performances in terms of compression ratio. PVQ achieves on the images we considered an average compression of 3:1, 38.25% better than bzip2 (applied on the plane–interleaved images). The table also shows the compression ratio in the near–lossless scenario, i.e., when a small quantization error is allowed. In this setup it is possible to reach 4:1, 4.77:1 and 5.46:1 for an error $K = \pm 1, \pm 2$ and ± 3 respectively. In the last column we report, as a reference, the compression ratio when only the indices are encoded and the quantization error is discarded. Table II shows, for the same parameters, the Signal to Quantization Noise Ratio:

$$SQNR\ (dB) = \frac{10}{D} \sum_{i=0}^{D-1} \log_{10} \left(\frac{\sigma_{I_i}^2}{\sigma_{E_i}^2} \right).$$

The partition boundaries for each of the five hyperspectral images are depicted in Figure 5. While similarities exist, the algorithm converges to different optimal boundaries on different input images. This is evidence that PVQ adapts the partitions to input statistics. Experimentally we have found that adaptation is fairly quick and boundaries converge to their definitive values in less than one hundred iterations.

8. CONCLUSIONS

We presented an extension of the GLA algorithm to the design of a Partitioned Vector Quantizer. The proposed algorithm is intended for the encoding of source vectors drawn from a high dimensional source on $\mathbb{R}^D$. PVQ breaks down the input space into $N \ll D$ independent subspaces and for each subspace designs a minimal distortion vector quantizer. The partition is adaptively determined while building the quantizers in order to minimize the total distortion. Experimental results on lossless and near–lossless compression of hyperspectral imagery are presented and discussed. Aside from competitive compression figures and progressive decoding, PVQ has a natural parallel implementation and it can also be used to implement search, analysis and classification in the compressed data stream.

Compression Ratio	Bzip2	Partitioned VQ				
	Lossless		Near–Lossless			Indices only
			K=1	K=2	K=3	
Cuprite	2.25	3.13	4.23	5.08	5.85	40.44
Low Altitude	2.13	2.89	3.88	4.66	5.34	39.10
Lunar Lake	2.30	3.23	4.22	5.03	5.78	47.03
Moffett Field	2.10	2.94	3.97	4.74	5.40	40.92
Jasper Ridge	2.05	2.82	3.70	4.36	4.92	35.02
Average	2.17	3.00	4.00	4.77	5.46	40.50

Table I: Lossless and near–lossless compression ratio.

SQNR (in dB)	Bzip2	Partitioned VQ				
	Lossless		Near–Lossless			Indices only
			K=1	K=2	K=3	
Cuprite	NA	NA	44.83	40.08	37.15	23.91
Low Altitude	NA	NA	47.94	43.18	40.19	25.48
Lunar Lake	NA	NA	48.53	43.75	40.79	27.15
Moffett Field	NA	NA	49.36	44.64	41.75	25.74
Jasper Ridge	NA	NA	46.94	42.19	39.24	20.37
Average	NA	NA	47.52	42.77	39.82	24.53

Table II: Near–lossless signal to quantization noise ratio.

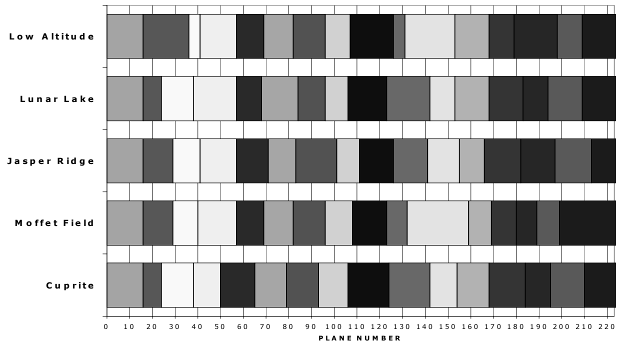

Figure 5: Partition sizes and alignment.

9. REFERENCES

[1] G. Shaw and D. Manolakis, "Signal Procesing for Hyperspectral Image Exploitation," *IEEE Signal Processing Magazine*, Jan 2002, Vol.19, N.1.

[2] B. Aiazzi, L. Alparone and S. Baronti, "Near–Lossless Compression of 3–D Optical Data," *IEEE Transactions on Geoscience and Remote Sensing*, Nov 2001, Vol.39, N.11.

[3] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Press, 1992.

[4] Y. Linde, A. Buzo and R. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, Jan 1980, Vol.28.

[5] http://popo.jpl.nasa.gov/html/aviris.freedata.html