High Performance Compression of Hyperspectral Imagery with Reduced Search Complexity in the Compressed Domain

Francesco Rizzo, Bruno Carpentieri
Dipartimento di Informatica e Applicazioni "R. M. Capocelli",
Università degli Studi di Salerno, Baronissi (SA) 84081 Italy

Giovanni Motta, James A. Storer
Computer Science Department, Brandeis University, Waltham, MA 02454 USA

Abstract: In previous work we considered LPVQ, a compression algorithm based on Locally Optimal Partitioned Vector Quantization that can be used to compress hyperspectral images by applying partitioned VQ to the spectral signatures (e.g., to the 224 16-bit values of a NASA AVIRIS pixel) and then encoding error information with a threshold that can be varied from high quality lossy to near lossless to lossless (e.g., 50-to-1 lossy, 10-to-1 near lossless, or 3-to-1 lossless). An advantage of LPVQ is extremely fast decoding (table lookup followed by entropy decoding), but it is at the cost of more complex encoding. Here we present a new low complexity algorithm for hyperspectral image compression, called SLSQ, that employs linear prediction targeted at spectral correlation followed by entropy coding of the prediction error. We then consider how SLSQ can be combined with LPVQ in a scenario commonly arising in practice. In this scenario, a low-complexity lossless encoder on the remote acquisition platform compresses the data for transmission to a central computing facility, where it is processed and re-coded using LPVQ, so that the compressed data can be distributed to the final users at various quality levels. The VQ indices of the LPVQ form a lossy compressed image of only about 2% of the original size; this small image can be employed to greatly reduce the time for browsing and classification.

1. Introduction

Air-borne and space-borne remote acquisition of high definition electro-optic images has been increasingly used in military and civilian applications to recognize objects and classify materials on the earth’s surface. By analyzing the spectrum of the reflected light it is possible to recognize material(s) composing an observed scene. Hyperspectral detector technologies have made possible the recording a large number of spectral bands over the visible and reflected infrared region. These instruments have sufficient spectral resolution to allow very accurate characterization of the spectral reflectance curve of a given spatial area. For example, images acquired with the NASA JPL’s Airborne Visible/Infrared Imaging Spectrometer (AVIRIS [2003]) have pixels covering an area of approximately 20x20 meters, with reflectance decomposed into 224 narrow bands, approximately 10nm wide each, in the range 400 to 2,500nm; spectral components are represented with a 16 bits precision.

Hyperspectral imagery is a rapidly growing source of remote sensed data, even though its precision pales compared with the millions of channels of a truly high resolution lab spectrometer. Higher resolution, space-borne spectrometers may be available in the future. In fact, increasing the number of bands, i.e. the spectral resolution, allows for more sophisticated analysis and increases the data rate by only a linear amount. The problem is that the acquisition of these images already produces large amounts of highly correlated data (e.g., in the range 140MB to 1GB for AVIRIS images) in the form of a two dimensional image matrix each pixel consisting of many components, one for each spectral band (e.g., 224 16-bit values for a pixel in an AVIRIS image). Since
hyperspectral imagery is acquired at significant cost and often used in tasks like classification or target detection, compression algorithms that provide lossless or near-lossless quality may be required. In addition, it may be desirable to have low complexity that allows efficient on-board implementation with limited hardware.

Traditional approaches to the compression of hyperspectral imagery are based on differential prediction via DPCM (Abousleman, Lam, and Karam [2002], Aiazzi, Alparone, and Baronti [2001], Abousleman [1995]), vector quantization (Mielikäinen and Toivanen [2002], Pickering and Ryan [2001], Manohar and Tilton [2000], Ryan and Arnold [1997]), or dimensionality reduction through principal component analysis (Sharp [2002], Subramanian, Gat, Ratcliff and Eismann [1992]). An inter-band linear prediction approach based on least square optimization is presented in Mielikäinen, Kaarna, and Toivanen [2002]; this compression method optimizes, for each sample, the parameters of a linear predictor with spatial and spectral support.

In previous work (Motta, Rizzo, and Storer [2003]) we presented a locally optimal partitioned vector quantizer (LPVQ) for encoding high dimensional data, and applied it to lossless, near-lossless, and lossy compression of AVIRIS hyperspectral images. The compression achieved by LPVQ is aligned with the current state of the art, and its decoding complexity is extremely low (entropy decoding followed by table lookup). However, encoding is relatively complex, and still more complex than decoding even when codebooks are supplied. Here we present a new low complexity algorithm for hyperspectral image compression, called SLSQ, that employs linear prediction along with least squares optimization, followed by entropy coding of the prediction error. We then consider how SLSQ can be combined with LPVQ to model a scenario commonly arising in practice, where a relatively weak encoder (e.g., an airborne remote sensor) employs SLSQ to compress data, compressed data is received by (or delivered to) a powerful central computing facility where the data is processed and re-coded using LPVQ. The LPVQ compressed data is browsed and downloaded by remote users (Figure 1).

![Figure 1](image)

**Figure 1:** A relatively weak collector sends data to a powerful central server, from where it is distributed to the final users.

The VQ indices of LPVQ account for only about 2% of the original size of the data. If the centroids in each codebook are sorted according to their energy, the \( N \) index planes retain important information on the structure of the original image. These planes alone can be used to browse and select areas of interest and, as we show here, to extract information that can be useful to speed up the classification of the original spectral signatures.

Leveraging this fact, the LPVQ indices are broadcasted to the end users for fast browsing and classification. At any point, they can request the enhancement layer and refine classification by resolving ambiguities. It is also the case that when the centroids are
sorted according to their energy, a single plane of indices (from any one of the VQ partitions) forms a visually pleasing grayscale image of the scene in question, which is also useful for browsing operations by humans. In a typical implementation, there are 16 of these planes, each only about 0.15% of the original size of the data.

Section 2 begins by presenting a low complexity hyperspectral compression algorithm based on inter-band prediction, and then presents the SLSQ algorithm, a more aggressive method that optimizes the predictor for each pixel and each band. Section 3 describes experiments with these predictive compression methods. Section 4 presents a fast classification that can be used on LPVQ encoded images; in the same section experiments showing roughly a factor of 10 speed-up are also described. Section 5 discusses future research.

2. Inter-band Prediction using Least Squares Optimization

Remote sensed images, like AVIRIS, show two forms of correlation: spatial (the same material tends to be present in many adjacent pixels) and spectral (one band can be fully or partly predicted from other bands). The spectral correlation is generally much stronger than the spatial correlation. Furthermore, dynamic range and noise levels (instrument noise, reflection interference, aircraft movements, etc.) of AVIRIS data are higher than those in photographic images. For these reasons the spatial predictor of LOCO-I tends to fail on this kind of data. For example, Figure 2 shows the performance in terms of bit per sample of the LOCO-I predictor versus LPVQ.

![Figure 2: Empirical entropy per band of JPEG-LS predictor vs. LPVQ.](image)

The median predictor of JPEG-LS is inefficient almost everywhere, and especially in the visible part of the spectrum that is characterized by large dynamic ranges and accounts for almost half of the data. Nevertheless, JPEG-LS fast and efficient compression would be highly desirable to an on-board, hardware implementation. Motivated by these considerations, we employ a linear predictor in the style of JPEG-LS for bands marked for intra-band coding (IB set) and a new inter-band predictor for the rest. The predictor (which we refer to as LP) uses a simple heuristic to detect contexts in which it is likely to fail. In such cases the prediction is corrected using information about the behavior of the inter-band predictor in the previous two bands. Both inter and intra band predictors rely on a causal data subset, \( \hat{x}_{i,j,k} \) of the pixel \( x_{i,j,k} \) in the \( i \)-th row, \( j \)-th column of the \( k \)-th band, as shown in Figures 3a and 3b. After the prediction step, the prediction error is computed and entropy coded with a simple arithmetic coder.
In Rinaldo [2002], as shown in Figures 4a and 4b, the optimal approach for the current pixel (given a reference plane and a 3D subset of past data, an optimal linear predictor, in the least square sense, is determined for each sample). We employ a prediction structure and notation similar to Brunello, Calvagno, Mian, and Rinaldo [2002], as shown in Figures 4a and 4b.

Although LP performs well in practice, it does not perform quite as well as a similar approach proposed by Mielikäinen, Kaarna, and Toivanen [2002], which does not make the inter/intra band distinction and uses a slightly different entropy coding. To improve upon the performance of LP, we use more aggressive method that optimizes the predictor for each pixel and for each band (given a reference plane and a 3D subset of past data, an optimal linear predictor, in the least square sense, is determined for each sample). We employ a prediction structure and notation similar to Brunello, Calvagno, Mian, and Rinaldo [2002], as shown in Figures 4a and 4b.

In the following, by $x(i)$ we denote the $i$-th pixel in the above enumeration of the 2D context of $x_{m,n,k}$. Moreover, $x(i,j)$ denotes the $j$-th pixel in the 3D context of $x(i)$. The $N$-th order prediction of the current pixel ($x_{m,n,k} \equiv x(0,0)$, we drop the subscript and the parenthesis when referring to the current pixel) is computed as:

$$\hat{x}_{m,n,k} = \hat{x}(0,0) = \sum_{j=1}^{N} \alpha_j \cdot x(0, j)$$
\[ d_{2D}(x_{m,n,k}, x_{p,q,k}) = \sqrt{(m-p)^2 + (n-q)^2} \]

\[ d_{3D}(x_{m,n,l}, x_{p,q,l}) = \begin{cases} 
\sqrt{(m-p)^2 + (n-q)^2} & \text{if } j=i \\
\frac{1}{4} \sqrt{(m-p)^2 + (n-q)^2} & \text{if } j=i+1 
\end{cases} \]

**Figure 4a:** 3DLSQ, 2D and 3D distance functions.

**Figure 4b:** 3DLSQ, 2D and 3D contexts and pixel enumerations based on \( d_{2D} \) and \( d_{3D} \) respectively from the current pixel.

The coefficients \( \alpha_0 = [\alpha_1, \ldots, \alpha_N]^T \) minimizing the energy of the prediction error

\[ P = \sum_{i=1}^{M} (x(i,0) - \hat{x}(i,0))^2 \]

are calculated using the well-known theory on optimal linear prediction. Notice that the data used in the prediction are a causal, finite sub-set of the past data and no side information needs to be sent to the decoder.

Using matrix notation, we write \( P = (C \alpha - X)^T \cdot (C \alpha - X) \), where

\[ C = \begin{bmatrix} 
x(1,1) & \cdots & x(1,N) \\
x(2,1) & \cdots & x(2,N) \\
\vdots & \ddots & \vdots \\
x(M,1) & \cdots & x(M,N)
\end{bmatrix} \quad \quad X = \begin{bmatrix} 
x(1,0) \\
x(2,0) \\
\vdots \\
x(M,0)
\end{bmatrix} \]

By taking the derivative with respect to \( \alpha \) and setting it to zero, the optimal predictor coefficients are the solution of the following linear system

\[(C^T C) \cdot \alpha_0 = C^T X.\]

Once the optimal predictor coefficients for the current sample have been determined, the prediction error \( \epsilon = \|x - \hat{x}\| \) is encoded similarly to the previous two methods. We apply this approach (here named SLSQ) directly to the 224 AVIRIS bands. Setting \( M = 4 \) and \( N = 1 \) allows to obtain an excellent tradeoff between computational burden and compression performance.
3. Experiments with Compression by Linear Prediction

Table 1 reports the compression ratio obtained by LP and SLSQ on the five "standard", publicly available AVIRIS images. We compare these results to JPEG-LS, JPEG2000 (Taubman and Marcellin [2001]), and LPVQ. A similar average compression of 3.06 to 1 is also reported in Mielikäinen, Kaarna, and Toivanen [2002], although the experiments reported there refer to a subset of our data set and to images having non standard dimensions. The baseline LP algorithm has been applied with \( IB = \{1,2,\ldots,8\} \) and no prediction threshold \( (T = \infty) \).

Results of improvements of baseline LP and SLSQ are reported in Table 2, where:

- **LP-CTX** uses a simplified version of the context modeling described in Weinberger, Seroussi, and Sapiro [2000]. The prediction threshold \( T \) is here set to 80.

- In the **HEU** option the \( IB \) set, common to all input images, has been determined by an off-line heuristic: for each scene of the data set we check which band is better compressed spatially (LOCO-I) rather than spectrally (LP/SLSQ). Bands that are encoded in inter-band mode at least 60% of the time form the \( IB \) set for the HEU mode.

- **OPT** assumes that the encoder checks for the best method first. This requires virtually no side information (1 bit/band) and a one band look-ahead.

LP-CTX with one band look-ahead improves LP by more than 2%, and matches LPVQ compression at the cost of a small increase of storage requirements over baseline LP. The SLSQ-OPT method outperforms LPVQ by more than 4% at a computational cost roughly double with respect to LP, but still well below the high computational cost of the design stage of LPVQ.

<table>
<thead>
<tr>
<th></th>
<th>JPEG-LS</th>
<th>JPEG2000</th>
<th>LPVQ</th>
<th>LP</th>
<th>SLSQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cuprite</td>
<td>2.09</td>
<td>1.91</td>
<td>3.18</td>
<td>3.03</td>
<td>3.15</td>
</tr>
<tr>
<td>Jasper Ridge</td>
<td>2.00</td>
<td>1.80</td>
<td>2.88</td>
<td>2.94</td>
<td>3.15</td>
</tr>
<tr>
<td>Low Altitude</td>
<td>2.14</td>
<td>1.96</td>
<td>2.94</td>
<td>2.76</td>
<td>2.98</td>
</tr>
<tr>
<td>Lunar Lake</td>
<td>1.99</td>
<td>1.82</td>
<td>3.28</td>
<td>3.05</td>
<td>3.15</td>
</tr>
<tr>
<td>Moffett Field</td>
<td>1.91</td>
<td>1.78</td>
<td>3.00</td>
<td>2.88</td>
<td>3.14</td>
</tr>
<tr>
<td><strong>AVERAGE</strong></td>
<td><strong>2.03</strong></td>
<td><strong>1.85</strong></td>
<td><strong>3.06</strong></td>
<td><strong>2.93</strong></td>
<td><strong>3.12</strong></td>
</tr>
</tbody>
</table>

Table 1: Comparison of compression achieved.

<table>
<thead>
<tr>
<th></th>
<th>LP-CTX</th>
<th>SLSQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BASE</td>
<td>HEU</td>
</tr>
<tr>
<td>Cuprite</td>
<td>3.04</td>
<td>3.07</td>
</tr>
<tr>
<td>Jasper Ridge</td>
<td>2.96</td>
<td>2.98</td>
</tr>
<tr>
<td>Low Altitude</td>
<td>2.79</td>
<td>2.79</td>
</tr>
<tr>
<td>Lunar Lake</td>
<td>3.06</td>
<td>3.08</td>
</tr>
<tr>
<td>Moffett Field</td>
<td>2.93</td>
<td>2.94</td>
</tr>
<tr>
<td><strong>AVERAGE</strong></td>
<td><strong>2.96</strong></td>
<td><strong>2.97</strong></td>
</tr>
</tbody>
</table>

Table 2: Improvements of baseline LP and SLSQ algorithms.
4. Fast Classification

Once data is received by a central server, it can be re-compressed using LPVQ, so that decoding by remote users is extremely fast. In addition, different levels of quality can be provided to remote users. Another advantage of re-coding with LPVQ is that the quantization indices provide valuable information for browsing and classification of the compressed data.

A classification task is defined by giving a target vector \( T = [t_0, t_1, ..., t_{D-2}, t_{D-1}] \), a distortion measure \( D(I,T) \) and a classification threshold \( \Delta \). We wish to determine, for every pixel \( I(x,y) \), whether \( D(I(x,y),T) < \Delta \). When this condition is satisfied, we say that \( I(x,y) \) belongs to the class \( T \).

If an image is lossy compressed, the coding error may compromise the result of the classification. In order to assess the classification accuracy on images lossy compressed by LPVQ, we have experimented with a number of standard images for various errors and threshold parameters. It is important to notice that, while the lossy results presented here are peculiar of LPVQ, the considerations on the near lossless modes apply to any compression scheme bounding the maximum absolute error on each vector component.

The results for three NASA images, Cuprite, Low Altitude, and Moffet Field (AVIRIS [2003]), are presented here. The absolute error introduced by LPVQ in near lossless mode is respectively bounded to a maximum of 1, 2, 4 and 8 units. The classification uses Euclidean Distance with a threshold \( \Delta \) in the range 1 to 50; typically, values in the range 5 to 35 are of interest in practical applications.

Figures 5, 6, and 7 show the percentage of errors in the classification of 8 targets selected from the NASA Cuprite, Low Altitude, and Moffet Field images. The 8 targets are hand picked by browsing the reduced resolution image formed by the index planes and are meant to represent targets of common interest (buildings, roads, water, vegetation, etc.). The classification of the original image is used as a reference and the error percentage reflects the sum of all false positive (not classified in the original), false negative (classified in the original and not classified in the lossy) and misclassifications (different clusters in original and lossy images). One can see from these figures how in near lossless, the classification is fundamentally unaffected; even with an error of \( \pm 4 \), the percentage of pixels erroneously classified is, in the worst case, below 0.06%. An error of \( \pm 8 \) still provides a classification accuracy higher than 99.86%. This accuracy is sufficient for most applications and smaller coding errors, achieve even lower percentages.

Figure 8 presents the classification errors for the lossy compressed images. With the compression parameters used in Motta, Rizzo, and Storer [2003] (i.e. 256 code vectors and 16 partitions) the indices and the code vectors alone achieve a classification error always lower than 3%. Even if this error rate may be unsuitable to some applications, it is still possible to take advantage of this characteristic to design a faster classifier. The lossy image is reconstructed from the VQ indices and code vectors. Figure 8 shows that this information alone is sufficient to classify most pixels exactly. By bounding the minimum and maximum quantization error achieved for each component and for each cluster, we can detect which pixels are likely to be misclassified. For these pixels only, it will be necessary to download and decode the error \( E(x,y) \) and run the classification on the fully reconstructed spectral signature. Bounds on the quantization error are available at design time or can be inferred from the encoded image.
Since the number of partitions and centroids are typically small, the bounds for each partition and centroid can be precomputed and arranged into a lookup table. From this table, indexed by the quantization indices of a pixel, we can determine the lower and upper bounds on the distortion $D(I(x,y),T)$, namely $D_{\min}(I(x,y),T)$ and $D_{\max}(I(x,y),T)$.

Since $I(x,y)$ belongs to the class $T$ only if $D(I(x,y),T) \leq \Delta$, we can observe that:

- If $D_{\min}(I(x,y),T) \geq \Delta$, the pixel $I(x,y)$ clearly does not belong to the class $T$;
- If $D_{\max}(I(x,y),T) < \Delta$, the pixel $I(x,y)$ trivially belongs to the class $T$;
- If $D_{\min}(I(x,y),T) < \Delta \leq D_{\max}(I(x,y),T)$ then the quantization indices and the bounds on the error are not sufficient to decide the classification. In this case, and only in this case, it is necessary to access the error vector, reconstruct the spectral signature $I(x,y)$ and compute $D(I(x,y),T) < \Delta$.

We are making the assumption that the quantities $D_{\min}(I(x,y),T)$ and $D_{\max}(I(x,y),T)$ can be computed by summing the individual contributions of each vector component $k$, upper and lower bounded by the minimum and maximum of the quantization residuals $\min(E_{k,i})$ and $\max(E_{k,i})$. This is clearly the case of Euclidean Distance, a distortion measure commonly used in the classification of hyperspectral images.

If, for a given pixel $I(x,y)$ we have that $D_{\min}(I(x,y),T) \geq \Delta$ or $D_{\max}(I(x,y),T) < \Delta$, then the pixel is classified by its indices, with a number of operations proportional to the number
of partitions $N$. Otherwise, the error vector $E(x,y)$ must be retrieved and $D(I(x,y), T) < \Delta$ computed explicitly, with a number of operations proportional to the number of bands $D$ (for AVIRIS images, $D = 224$).

The speed of the classification depends on the probability:

$$\Pr(D_{\min}(I(x,y), T) < \Delta \leq D_{\max}(I(x,y), T))$$

that the two test fail and that we have to use the error vector in order to classify. In particular, if $\tau_{\text{sub}}$, $\tau_{\text{add}}$, and $\tau_{\text{mul}}$ are respectively the times to perform a subtraction, a sum, and a multiplication on a given computing platform, the running time is given by:

$$2(N-1)\tau_{\text{add}} + (D\tau_{\text{sub}} + \tau_{\text{mul}} + (2D - 1)\tau_{\text{add}}) \cdot \Pr(D_{\min}(I(x,y), T) < \Delta \leq D_{\max}(I(x,y), T))$$

Figure 9 shows an experimental assessment of the probability $\Pr(D_{\min}(I(x,y), T) < \Delta \leq D_{\max}(I(x,y), T))$. Three AVIRIS images have been classified with respect 100 targets randomly selected. The figure averages the results obtained for thresholds $\Delta$ in the range 5 to 35; the curves represent three quantizers having 256, 512 and 1024 levels each. The percentage of vectors on which the simplified test fails to classify the points reaches a maximum between 7.5% and 11% for a 1024 levels quantizer and 12% to 18% for a 256 level quantizer. Practical figures are typically better. Problems like target detection are solved by applying a low threshold, while classification requires a high one; in these regions of the curve, the percentages are typically lower.
5. Future Research

The proposed methods for compression by linear prediction depend loosely on the entropy coder; experiments are in progress to assess the effect of replacing the arithmetic coder with the CCSDS standard algorithm (CCSDS [1997]) for lossless data compression for space applications (a hardware implementation is widely used on many satellites). We are also conducting further experiments to use linear prediction and least square optimization schemes to improve LPVQ lossy compression (that is, compression of the indices), with positive repercussions on the lossless part as well. Our classification results can be used in two ways. Given an image encoded with LPVQ, we can improve the naïve classification with the use of a lookup table, or alternatively, we can use this as a tool to select the number of quantization levels to match a desired average classification speed. A more precise understanding of this sort of "tuning" is being investigated.

References


