

# A lossless compression algorithm for hyperspectral data \*

I. Gladkova<sup>1</sup>, M. Grossberg<sup>1</sup>

<sup>1</sup> CCNY, NOAA/CREST, 138th Street and Convent Avenue, New York, NY 10031

## ABSTRACT

In this paper, which is an expository account of a lossless compression techniques that have been developed over the course of a sequence of papers and talks, we have sought to identify and bring out the key features of our approach to the efficient compression of hyperspectral satellite data. In particular we provide the motivation for using our approach, which combines the advantages of a clustering with linear modeling. We will also present a number of visualizations which help clarify why our approach is particularly effective on this dataset.

At each stage, our algorithm achieves an efficient grouping of the data points around a relatively small number of lines in a very large dimensional data space. The parametrization of these lines is very efficient, which leads to efficient descriptions of data points. Our method, which we are continuing to refine and tune, has to date yielded compression ratios that compare favorably with what is currently achievable by other approaches. Based on a data sample consisting of an entire day's worth of global AQUA-EOS AIRS Level 1A counts data (235 granules), our compression algorithm was empirically demonstrated to achieve a lossless compression ratio on the order of 3.7 to 1.

**Keywords:** compression, hyperspectral data, spectral clustering, multiple eigenspaces

## 1. INTRODUCTION

Hyperspectral imaging is becoming an increasingly important tool for monitoring the earth and its environment from spaceborne and airborne platforms. Hyperspectral imaging data consists of measurements from a scene which span both space and spectrum. These measurements differ from conventional imaging in that possibly thousands of spectral channels are measured. They differ from spectrometer measurements in that the measurements span both a spatial and often temporal extent.

The greatly expanded volume of data provided by these instruments presents tremendous challenges for transmission, computation, storage, distribution and archiving. The current recommendations from the international Consulting Committee for Satellite Data Standards (CCSDS) for data compression, 121.0-B-1 and 122.0-B-1, may be applied to hyperspectral data, but give low compression ratios. All the major space agencies across the world, including NASA, NOAA, CSA, ESA, CNES, and ASI have recognized that the current techniques are not adequate, hence novel methods optimized for hyperspectral imaging data must be employed.

Our approach to hyperspectral data compression exploits properties of hyperspectral imaging that are known a priori to achieve superior results over generic compression methods. There is an extensive literature on generic data and image compression.<sup>1</sup> All compression is based on implicitly or explicitly exploiting the statistical structure of the data. By accurate modeling and identification of properties special to hyperspectral imaging we are able to obtain a more compact representation of the data. The primary assumptions on the measurements are:

1. there are a small number of physical processes significant for generating the spectral radiances
2. the measured spectral radiances are highly correlated
3. the correlations in both the spatial and spectral domain have a piecewise smooth component
4. the measurements have a component which is independent and stochastic

With this in mind, we have developed a compression algorithm based on separating the measurements into a piecewise smooth and a stochastic component. The piecewise smooth component models the correlations between nearby measurements. The stochastic part is approximately independent between measurements within the set. Hence, our approach is to construct a model which fits a piecewise smooth low parameter description of the data, and subtracts it from the sample. If the fitting captures all the important dependencies, the residual values are independent. Empirically, the residual values are gaussian, and can be compressed optimally using entropy encoding.

---

\*Sponsored by NOAA/NESDIS under Roger Heymann (OSD), Tim Schmit (STAR) Compression Group

## 2. DATA

The compression algorithm is tested empirically using AIRS data. The AQUA-EOS AIRS is the first of the new generation of high spectral resolution infrared sounders. AIRS is an IR grating spectrometer that measures upwelling radiance between  $3.74\mu m$  and  $15.4\mu m$  in 2378 channels (e.g.,<sup>2</sup>). AIRS takes 90 measurements as it scans 48.95 degrees perpendicular to the satellite's orbit every 2.667 seconds. We use data granules made up of Level 1A digital counts representing 6 minutes (or 135 scans) of measurements. These data are processed operationally at the National Oceanic and Atmospheric Administration, National Environmental Satellite, Data and Information Service (NOAA/NESDIS).<sup>3</sup> The near real-time AIRS data offer a unique opportunity to use empirical satellite observations as a test-bed for data compression studies of future hyperspectral sounding instruments.

## 3. COMPRESSION FROM STRUCTURE IN MEASURED RADIANCES

The goal in hyperspectral imaging is to determine across space, time, and wavelengths, the distribution of radiances. These radiances partially reveal both properties and state of the scene by determining the reflectance and scattering of light from the sun, and the emission of light through thermal radiation (heat). Variations in reflectance, scattering, and thermal radiation arise from diverse phenomena such as the presence of clouds, the temperature profile across altitudes, as well as variations in water vapor and other gases (such as CO<sub>2</sub>, ozone, methane, etc), aerosols, and surface properties.

The observed variations in radiances come from two qualitatively different sources. One source of variation comes from the fact that the distribution of radiance sources and scatterers satisfies differential or integral equations. For example, in the thermal IR domain, radiances are determined by the distribution of temperatures in the atmosphere as mediated by radiative transfer equations. Solutions to such equations tend to be locally smooth almost everywhere with abrupt discontinuities along singularities. The other source of variations in radiances in the scene is stochastic in nature. For example distributions of individual clouds and the arrival of photons at the instrument, are well modeled as random processes.

In addition to the two sources of variations in the radiances, there are similarly two sources of variation that come from our measurement processes of the radiance. One source of variation tends to be locally smooth with possible changes at singularities, combined with random fluctuations. Locally smooth variations also come from biases in the instruments. These include spatial attenuation of the data due to varying polarization with changes in the scan geometry, and varying sensitivity of the basic sensor across the spectrum. The random fluctuation in measurement comes from electronic noise as well as from the thermal dynamics of the sensor.

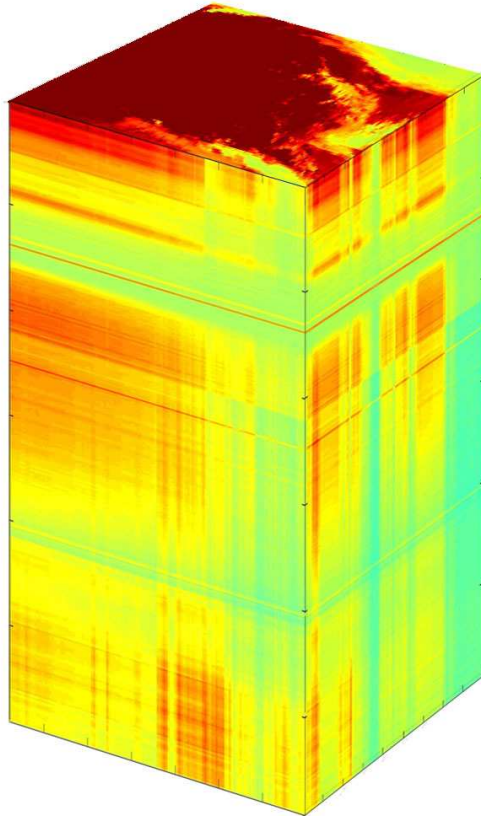
Fortunately, the aggregation of these effects combines so that the variation in the raw measurements can be well modeled as a sum of a piecewise smooth component and a zero-mean stochastic component. This is the basis of our compression method. The raw measurements are collected into arrays of data called feature vectors, which are treated independently. The measurements that make up these feature vectors are related both spatially and spectrally and chosen so as to capture all significant dependencies in the data. The piecewise locally smooth structure allows us to smoothly index these arrays compactly with a small number of parameters we will call local coordinates. It also allows us to determine a map from the local coordinates back to the data space in a way that approximates the full vector of measurements well. In this way, we can store the parameters of a map, and the low parameter local coordinates for each data point. This results in a large reduction in the amount of data that needs to be stored to approximate the original data.

The approximation can be used to achieve lossless compression by adding a subsequent step that preserves all information. To preserve all the information, the residual errors between the approximation and the data are stored. We should note in this connection that the measured data are stored in the integer form as discussed in the next section. Ideally the remaining residual component will only contain the stochastic or noise-like portion of the data. Since the errors are typically very small, the residual components have a much smaller average dynamic range, hence can be encoded with relatively few bits. In our approach, the statistical correlations between individual measurements are contained in a piecewise smooth manifold component. The residual component may be treated as independent noise and entropy encoded using a conventional compression technique, such as Huffman encoding,<sup>4</sup> which is optimal under these assumptions. While our model can be made arbitrarily flexible for the part of the data that is piecewise smooth, there are hard theoretical limits to the lossless compression of stochastic sampled data. Thus our separation focuses effort on that portion of the data which can be compressed while maintaining optimal compression on the more difficult portion.

#### 4. MEASUREMENT FORMAT

For the purposes of compression, it is most effective to work directly on the raw digital counts produced by the AIRS instrument. Information theory imposes certain limitations on the ability to compress data without losing information.<sup>5,6</sup> We have chosen to work with digital counts rather than post-processed a rounded version of the radiance data, since in line with the previous observation, this processing introduces additional randomness.

In the context of our work the AIRS granules are organized into 2378 infra-red (IR) spectral channels, for each footprint. We use 1501 stable channels selected by NOAA scientists. The granule is therefore made up  $K = 1501$  channels spectrally,  $I = 90$  footprints per scan line and  $J = 135$  scan-lines. Figure 1 shows a visualization of Channels 1207-1501 of an AIRS granule.



**Figure 1.** Visualization of a portion of a three dimensional AIRS hyperspectral data cube. Note the strong correlations between neighboring channels.

These  $I \times J \times K$  hyperspectral data cubes can be represented as a bounded integer-valued function  $q(x, y, \nu)$  defined on the integer lattice points  $1 \leq x \leq I, 1 \leq y \leq J, 1 \leq \nu \leq K$ . Here  $x, y$  indices can be interpreted as coordinates on the geospatial grid and the  $\nu$ -corresponds to the sensor's spectral channel. The values  $q$ , or digital counts, are a quantization of an analogue signal which is approximately linear in irradiance at the sensor, integrated over a narrow portion of the spectral domain. The precise relationship with scene radiance is known from calibration methods described in,<sup>7</sup> but is not crucial here since our compression method operates directly on digital counts. The quantization into digital counts, has a range of  $[0, 2^{N_{\text{bits}}} - 1]$ , where  $N_{\text{bits}}$  is the number of bits per measurement. For AIRS three ranges of channels were given different bit depths due to the varying noise characteristics. For the first 514 channels the bit depth was  $N_{\text{bits}} = 12$ , for spectral channels 515-1205,  $N_{\text{bits}} = 13$  and for 1206-1501,  $N_{\text{bits}} = 14$ .

## 5. COMPRESSION OF FOOTPRINT SPACE

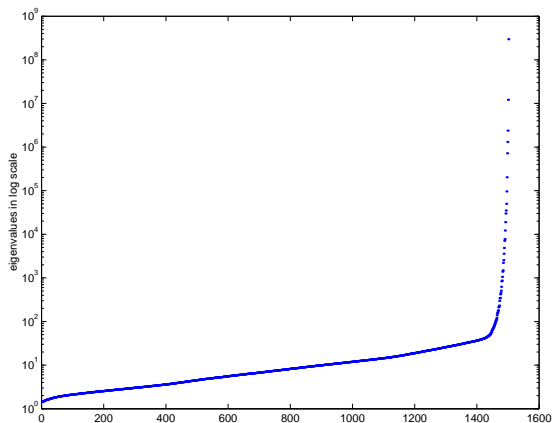
We can interpret a granule as a sequence of 1501 single channel geospatial images  $q(x, y, \nu)$  of size  $90 \times 135$ , each obtained by fixing the spectral parameter  $\nu$ . Alternately, we can consider the granule as a collection of vectors  $q_{x,y} = q(x, y, \nu)$  representing a quantized sampling of the spectral curve for each footprint. Our algorithm is based on the fact that the digital counts have a strong degree of coherence across spectral channels for a single footprint. This is motivated by the fact that the digital counts across channels depend on the temperatures and the water vapor content of the atmosphere at different altitudes. These quantities vary smoothly with some discontinuities. Hence, our compression approach is based on extracting the spectral redundancy by approximating the spectral curve in each footprint from a family of curves with a small number of parameters.

Computationally, it is more convenient to work with the set of vectors  $q_{x,y}$  for all the footprints. Our goal will be to represent these values in an efficient manner. The set of footprints are contained in a  $K$ -dimensional Euclidian space  $\mathbb{R}^K$ . For example,  $K = 1501$  for AIRS granules. In our implementation we have found it effective to decompose the spectral channels into disjoint dimension ranges and treat them independently, each with its own value for  $K$ . We denote all vectors  $q_{x,y} \in \mathbb{R}^K$ , for all footprints in the granule by  $\mathbf{Q}$ .

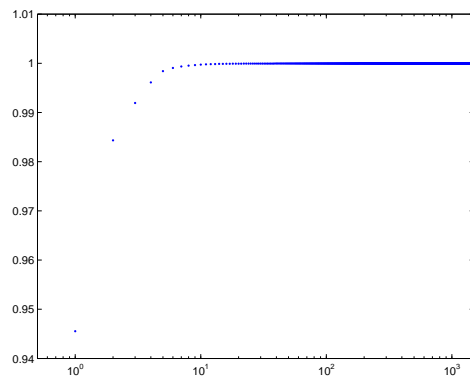
Lossy compression seeks to approximate elements of  $\mathbf{Q}$  compactly while minimizing the approximation error. In lossless compression we are required to reconstruct  $\mathbf{Q}$  without error. Nevertheless, a method which is able to approximate  $\mathbf{Q}$  with a small approximation error can also be used for lossless compression. Rather than store the original measurements, one stores the parameters for the approximation, and the approximation error values or residuals at each point. One error value must be stored for each measurement. Although the number of entries that must be stored is the same as the original number of measurements, we should note that if the approximation is good, the average range of values is quite small. This means the number of bits per measurement required can be greatly reduced using entropy encoding. Conversely a good approximation should be compact, requiring only a few parameters, and thus represents a significant reduction in size from the original data. Hence, a combination of the low parameter approximation and the low bit rate residual makes it possible to achieve high compression losslessly.

## 6. LINEAR APPROXIMATION OF FOOTPRINTS

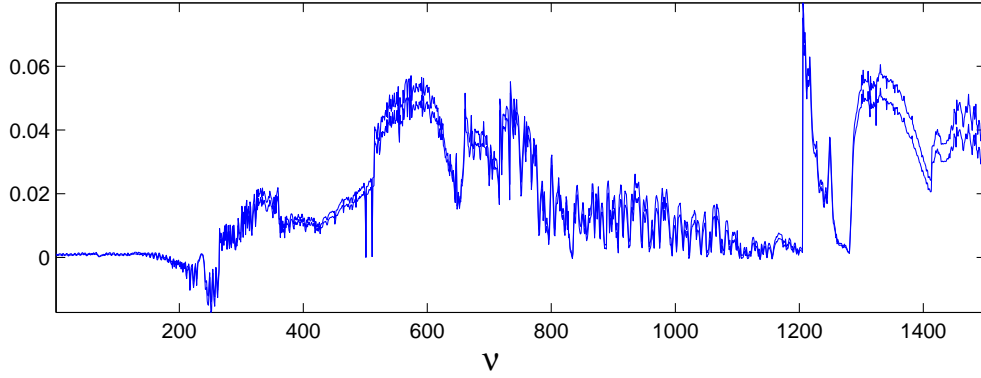
In hyperspectral image data the footprints occupy a high dimensional space  $\mathbb{R}^K$ . The effective degrees of freedom of scene radiance, and thus digital counts, should be significantly smaller than  $K$ . Empirically, the points in  $\mathbf{Q}$  tend to aggregate near considerably lower-dimensional hyperplane of  $\mathbb{R}^K$ . Note that the local number of degrees of freedom of this space will typically be considerably smaller than the apparent dimension due to strong spectral correlations. One strategy to find this plane is principal component analysis, also known as the Kohonen-Louve (KL) transform.



**Figure 2.** Rapidly decaying eigenvalues for the covariance of footprints in a granule.



**Figure 3.** Total cumulative energy explained by finite dimensional hyperplanes of footprint space. More than 99% of the energy is explained with only 3 components. Note, the graph is shown in log-scale for visual clarity.



**Figure 4.** The envelope of multiple estimates of the first principal component. The narrow width of the envelope indicates that a well defined direction of greatest variation in the data set.

Underlying the PCA algorithm is an assumption that the data has an approximately Gaussian distribution. For such a distribution, coordinates may be established which efficiently represent the data. A Gaussian is parameterized by a vector representing its center  $\mu$ , and a symmetric matrix  $\Sigma$ , which represents directions and relative variances of an orthogonal coordinate system aligned to the data. The vector  $\mu$  is estimated by computing the mean of the data and  $\Sigma$  is obtained by computing the covariances. The eigenvectors of  $\Sigma$  form an orthonormal basis of  $\mathbb{R}^K$ . The variance of the data along the direction of each of the eigenvectors is the corresponding eigenvalue. Hence the size of the eigenvalue determines how significant that dimension is in describing the variation of the data set. The graph in Figure 2 shows the eigenvalues for the covariance of footprints in a data set.

The rapid decay of the eigenvalues shows that it is possible to explain most of the variation in the footprint space with a few parameters. In Figure 3 we see that only 3 components are needed to approximate the data so that 99% of the variation is represented, as measured by a least squares error metric. Figure 4 shows multiple estimates of the eigenvector associated to the largest eigenvalue. This eigenvector is called the first principle component. Each of the estimates was obtained by independently choosing a collection of random samples. Note that the estimates are very consistent.

The line passing through the mean in the direction of the first principle component is the best least squares fit of a line to the data. Similarly the plane passing through the mean, spanned by the first two eigenvectors, is the best plane approximation in the least squares sense. If the data lives close to an  $N$ -dimensional hyperplane of  $\mathbb{R}^K$ , the smallest  $K - N$  eigenvalues will be very small. Hence, the data can be accurately approximated by projection onto the hyperplane  $V_N$ , passing through the mean and spanned by an first  $N$  eigenvectors vectors

$$\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N. \quad (1)$$

These are known as the principal ( $N$ ) components. The coordinates of a point  $\mathbf{q} \in \mathbf{Q}$  projected into the principal  $N$ -dimensional hyperplane is given by

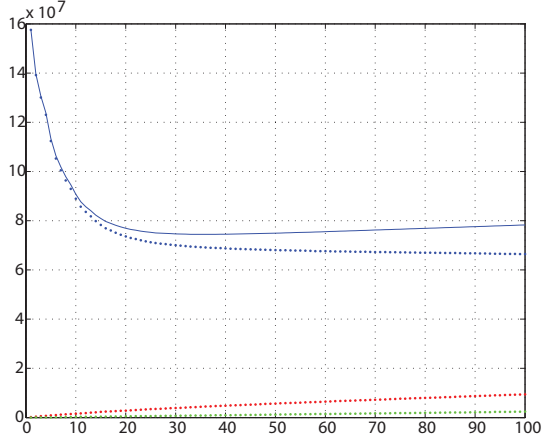
$$\rho_n = (\mathbf{a}_n, \mathbf{q} - \mu), \quad 1 \leq n \leq N. \quad (2)$$

The error vector

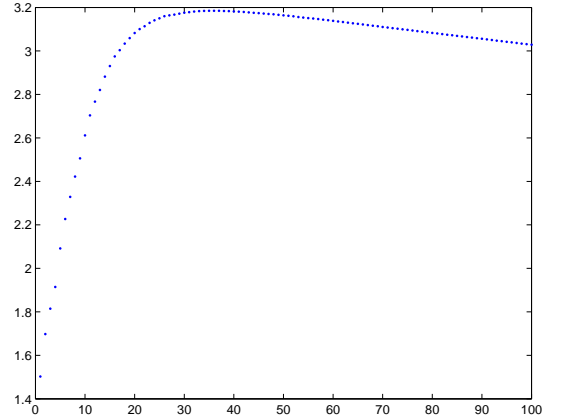
$$\tilde{\mathbf{q}}_\varepsilon = \mathbf{q} - \mu - \rho_1 \mathbf{a}_1 - \dots - \rho_N \mathbf{a}_N, \quad (3)$$

is a vector perpendicular to the hyperplane  $V_N$ .

The coefficients  $\{\rho_n\}_{n=1}^N$ , are indexed by  $(x, y)$ . These can thus be interpreted as  $N$  2D images. These coefficients, the vectors  $\mathbf{a}_n$ , and differences  $\tilde{\mathbf{q}}$  together make it possible to reconstruct the original file. The objective function we are



**Figure 5.** Memory requirements of storing the footprints as a PCA based representation as a function of the dimension of the approximating hyperplane.



**Figure 6.** Compression ratio as a function of number of PCA components.

trying to minimize is the size of the compressed file. PCA is, however minimizing the least squares Euclidian error to a hyperplane. This is a convenient proxy which in some cases is equivalent, see<sup>8</sup> for discussion.

## 7. RESIDUAL ENCODING

The PCA approximation has the effect of removing correlated variations in measurements across channels. The coordinates of the residual vector  $\widetilde{\mathbf{q}}_\varepsilon$  are variations which can be approximated by independent and identically distributed (iid) noise. This noise is the component of our data model that we refer to as stochastic. Our model of the data as a combination of a piecewise linear component and a stochastic component is of course a familiar signal plus noise model. Let  $\{\rho_n\}_{i=1}^N = \mathbf{X}$  be a vector-valued random variable. We will model a spectral set of measurements of a footprint as a vector-valued random variable  $\mathbf{Y}$  which satisfies the relation:

$$\mathbf{Y} = f(\mathbf{X}) + \eta, \quad (4)$$

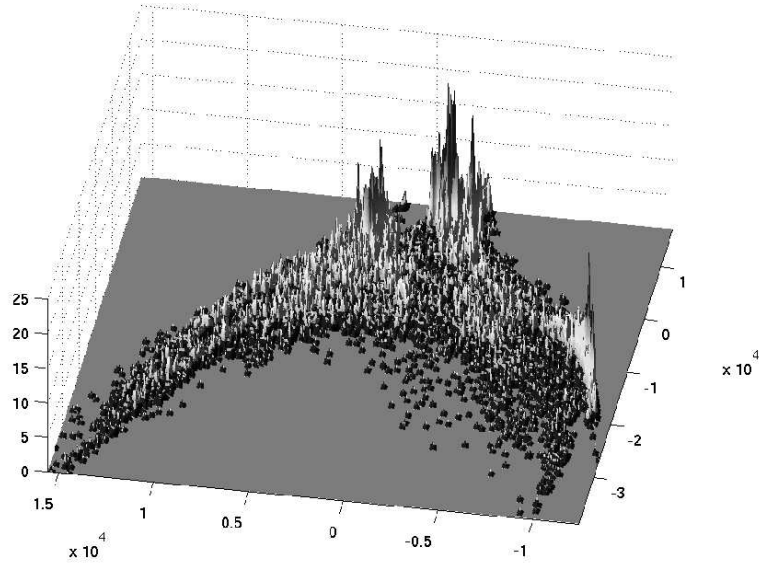
where the model  $f(\mathbf{X})$  is assumed to be linear, and  $\eta$  is a vector-valued random variable representing iid noise.

Huffman coding is one example of a coding scheme that is optimal for these assumptions. The compression achieved is a combination of the compressed residuals, the compressed PCA coefficients, and the PCA eigenvectors.

Figure 5 shows the memory sizes of these three data components as a function of the number of principal components. The memory size needed to store the residual errors (the blue dotted curve) monotonically decrease as more principal components are used. Initially the reduction in the size of the residuals is large compared with the extra space required to store more principle component coefficients (the red dotted line), as well as the principal components themselves (the green dotted line). At some point increasing the number of components simply exchanges memory required to store residuals for memory needed to store the principle component coefficients. However, as can be seen in Figure 6, in the interval between 30 and 40 dimensions the ratio curve flattens and then starts to decrease, due to additional overhead needed to store principle components.

## 8. PIECEWISE MODEL

While a PCA based linear model is capable of capturing correlations between different dimensions, it assumes that the data is clustered about a single hyperplane. To understand this better, consider an equal mixture of two sharply peaked gaussians in one dimension with widely separated means. If this mixture is treated as a single gaussian, PCA would estimate a large variance, and a  $\mu$  between the true means. Hence the residuals from Eq. (3) will be quite large, almost certainly limiting



**Figure 7.** A 2D Histogram of AIRS footprints projected onto the first two principle components. It is clear that that the footprints are not distributed as a single gaussian.

compression. If we construct a classifier that assigns data to the nearest cluster and store the cluster mean, the residuals could be dramatically reduced.

Can the AIRS data best be represented by a single or by multiple clusters? As a visual answer to this question we project footprints onto the first two principal components since they explain most of the variance as seen in Figure 2-4. In Figure 7, a 2D histogram shows the density of data points projected into these two principle components. At least three large clusters are apparent. This suggests that a successful compression algorithm should combine PCA and clustering. In particular, the approach is to cluster the data set and perform PCA on a per-cluster basis. If the number of clusters is relatively small, this can increase the flexibility of the model at a modest cost when compared to using more principle components. For each cluster we must store a set of eigenvectors and a mean. In terms of Figure 5 this increases the relatively small amount of memory used by the eigenvectors (the green dotted line). It is important to note that using multiple clusters does not increase the number of coefficients (the red dotted line) required and yet further reduces the memory required to store the residuals.

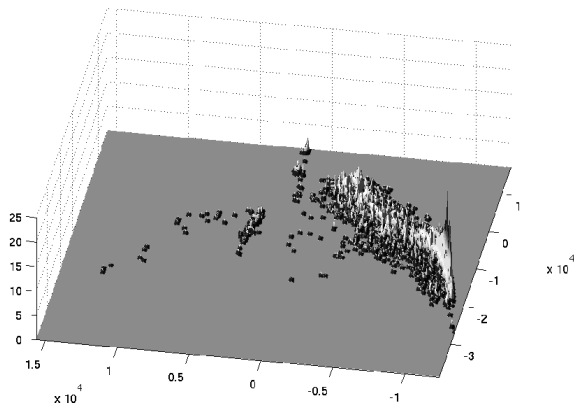
If we have  $M$  clusters, we can associate a matrix  $A_m$  to each cluster whose columns  $\mathbf{a}_{m,k}$  are eigenvectors of the covariance of that cluster. The error of this representation can be written as an objective function:

$$L(A_1, A_2, \dots, A_M) = \sum_{\mathbf{q} \in \mathbf{Q}} \min_{1 \leq m \leq M} \mathbf{d}^2(A_m, \mathbf{q}). \quad (5)$$

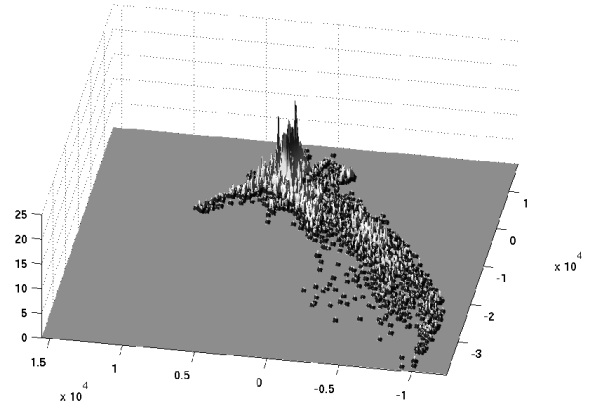
where  $\mathbf{d}(A_m, \mathbf{q})$  is the distance from a point  $\mathbf{q}$  to the hyperplane spanned by the columns of the matrix  $A_m$ . We will this denote this hyperplane of dimension  $N$  by  $V_m = A_m(\mathbb{R}^N) \subset \mathbb{R}^K$ .

The distance from a  $\mathbf{q}$  to the closest hyperplane is defined by the minimum of  $\mathbf{d}(A_m, \mathbf{q})$  with respect to the index  $m$  and the objective function in (5) is the sum of the squares of distances from every point  $\mathbf{q}$  in  $\mathbf{Q}$  to the closest hyperplane defined by  $A_m$ 's. If a point is found to be relatively close to more than one cluster, then it is assigned so that when projected onto the cluster's principle hyperplane, it is closest to the cluster mean (resolving ties). So let  $\mathbf{c}(\mathbf{q})$  be a map that describes the above-determined assignments for each point  $\mathbf{q}$ .

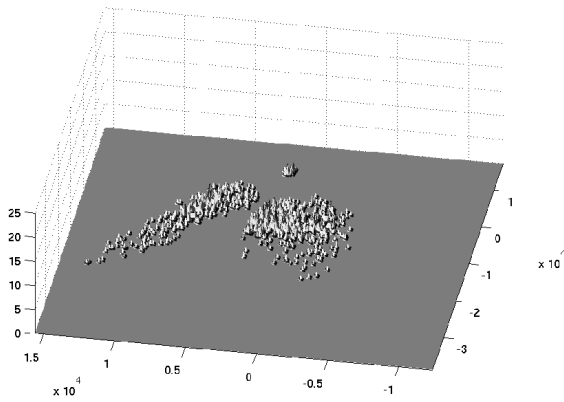
The optimization of the objective function  $L$  in (5) is done iteratively. At each step of the iteration we compute  $N$  first principal directions,  $\mathbf{a}_n$ , for each of the  $M$  current iteration's clusters, and the resulting distances  $\mathbf{d}(A_m, \mathbf{q})$  from each point  $\mathbf{q} \in \mathbf{Q}$  to  $M$  hyperplane  $\mathbb{R}^N(A_m)$  defined by  $\mathbf{a}_n$ ,  $n = 1, \dots, N$ . The smallest of the distances will determine the cluster to which that point belongs and the ties are resolved as mentioned above. After effecting the resulting reassignments, the



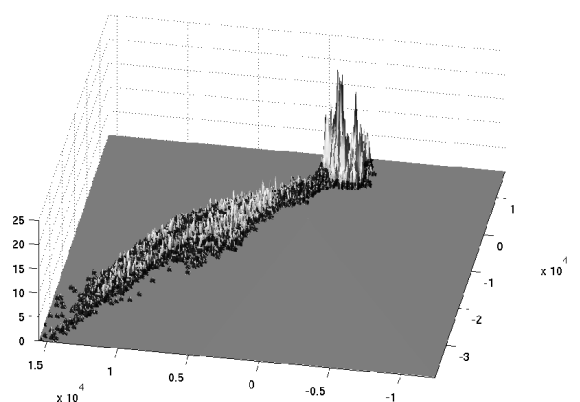
**Figure 8.** A 2D Histogram of the **first** cluster projected onto the first two principle components of the cluster.



**Figure 9.** A 2D Histogram of the **second** cluster projected onto the first two principle components of the cluster.



**Figure 10.** A 2D Histogram of the **third** cluster projected onto the first two principle components of the cluster.



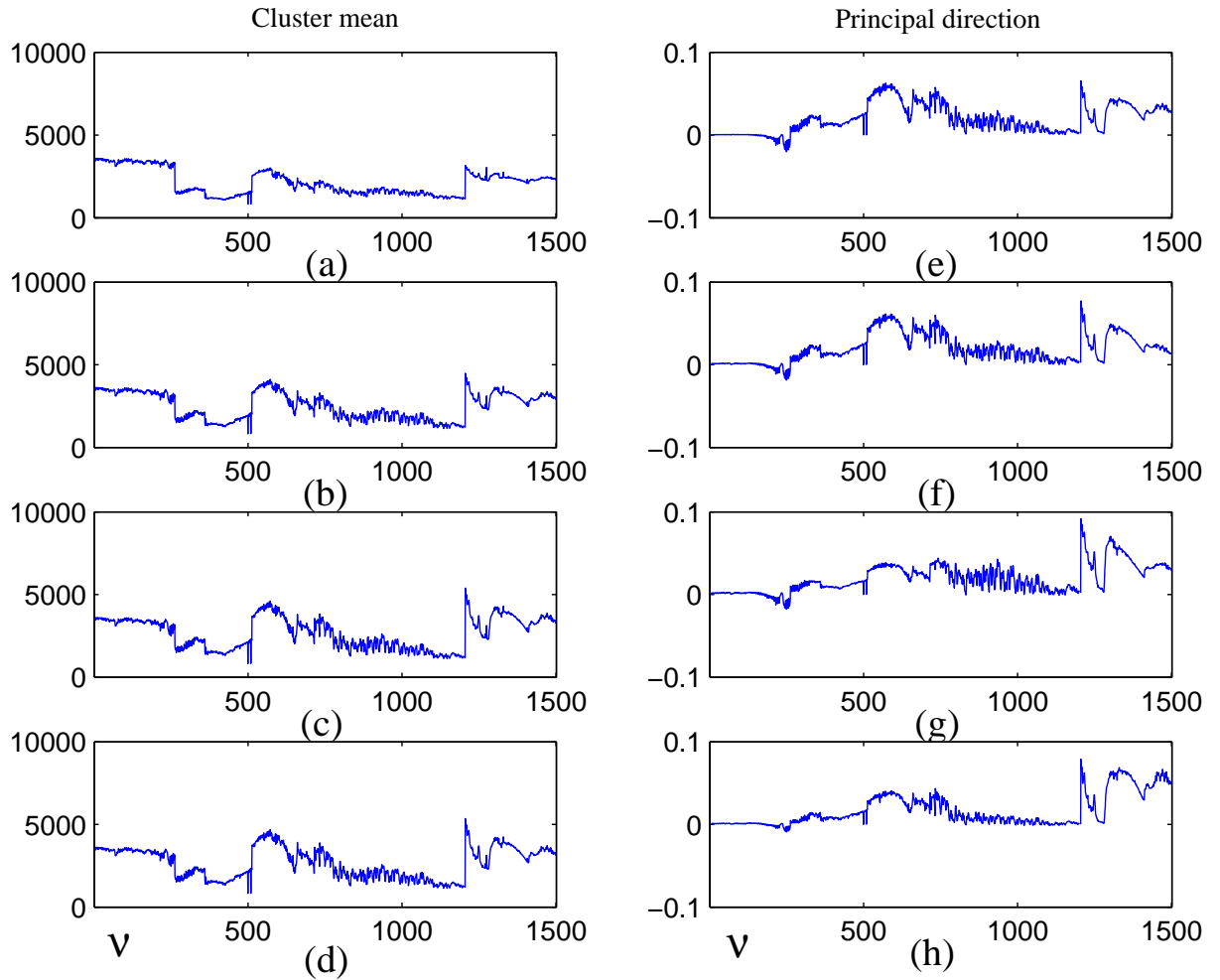
**Figure 11.** A 2D Histogram of the **fourth** cluster projected onto the first two principle components of the cluster.

next iteration of the algorithm begins. Note that this algorithm can be viewed as a version of the K-means cluster algorithm for hyperplane.<sup>9</sup>

As with most iterative algorithms, the initial conditions are very important. In order to find the global optimum, the initialization should begin reasonably close. There are a number of general methods that have been proposed to initialize *K*-means type problems and avoid local minima. For this data, we found that a rather simple approach is satisfactory. The initialization procedure begins by fitting a line *l* to the entire data set *Q*. Consider the projection of the dataset onto the line *l*. This line is broken up into segments, so that an approximately equal density of points projects onto each line segment. Since the principle direction is the direction that best explains the variance of the data, this segmentation results in an initial partitioning of the data into approximately equal parts.

In the following set of figures, we show an example of the clustering described above. Four clusters are chosen for visual clarity. The actual number of clusters depends on the granule and is determined adaptively. Figures 8- 11 display a 2D histogram for each cluster. Each histogram characterizes the density of data points associated with an individual cluster, projected into its own principle components. Figure 12 contains plots for the means and principal directions of each cluster. Note that means of the clusters 2 and 4 (plotted in Figure 12(a) and Figure 12(c)) appear to be very similar, but their principal directions (plotted in Figure 12(e) and Figure 12(g)) are not. The same phenomena can be observed in Figures 9 and 11, where the locations of the peaks of the corresponding histograms are very close, but the orientation of the principal directions for clusters 2 and 4 are quite different. It can also be observed that the orientation of clusters 1 and 2 (cf. Figures 8 and 9) are very similar, but the location of the peaks of these clusters are far apart. In this case, the principle





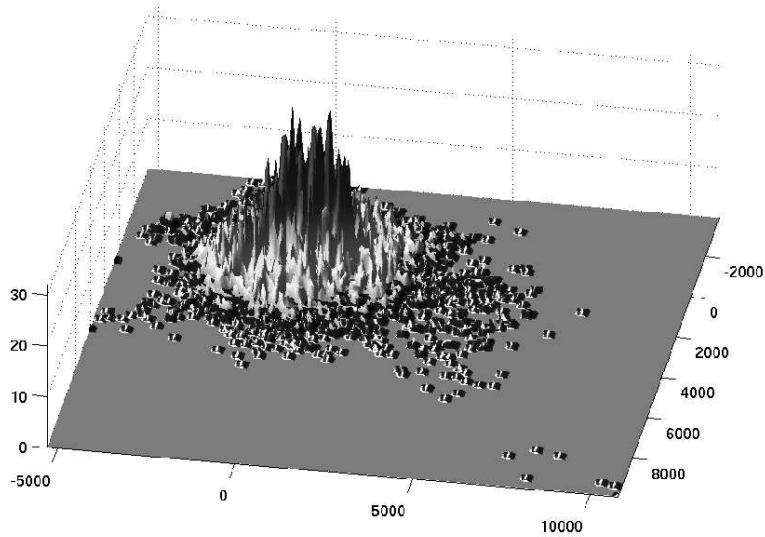
**Figure 12.** Means and principal directions for each of the four clusters.

directions are similar (cf. plots 12(e) and 12(f)), but the clusters have different means (cf. plots 12(a) and 12(b)).

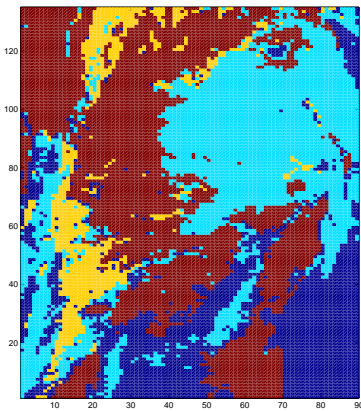
In Figure 14 we have colored each footprint location according to its assignment by  $c(\mathbf{q})$  to one of the four clusters. Note that although our algorithm operated in the footprint space without making use of spatial proximity, the assignment map indicates there are strong spatial dependencies between footprints in the granule.

After having subtracting from each footprint the projection onto corresponding first principal direction for corresponding to each cluster, we are left with a set of residuals. These residuals appear very close to gaussian as is apparent from Figure 13. It is also clear that the residuals are not isotropic. This makes it possible to reduce the size of the residuals still further by projecting and encoding the principal components of the residuals. The remaining residuals are encoded using Huffman encoding.<sup>4</sup>

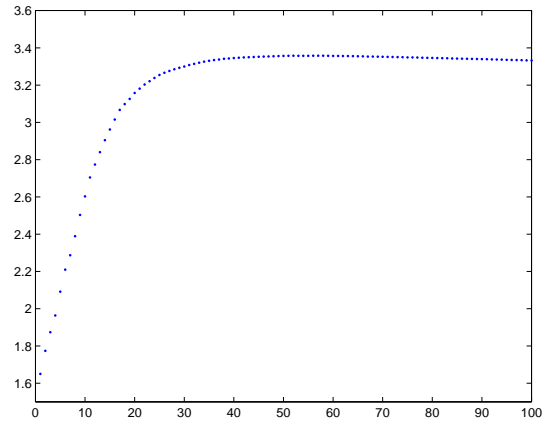
Compression ratio as a function of number of principal components for the demonstration case with just four clusters is shown in Figure 15. Note that the usage of four clusters has improved the compression ratio from 3.2 to 1 to 3.4 to 1. The results for adaptive clustering are given in the following section.



**Figure 13.** A 2D Histogram of remaining part after projecting onto 4 hyperspaces.



**Figure 14.** Assignment map.

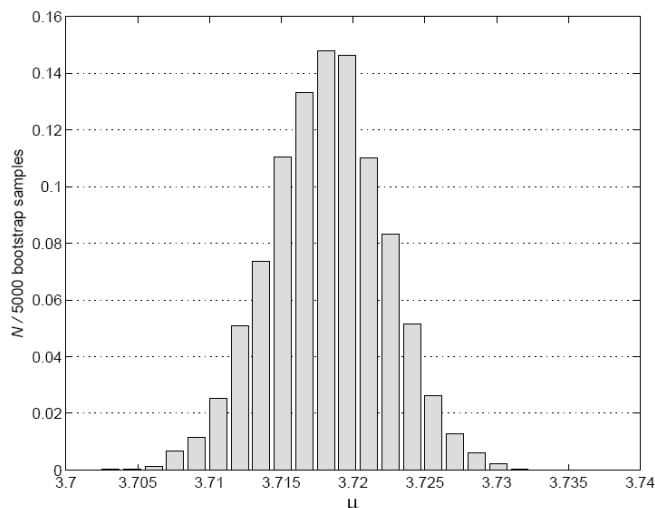


**Figure 15.** Compression ratio as a function of principal components for the considered example with just 4 clusters.

## 9. RESULTS

Based on a data sample consisting of an entire day's worth of global AQUA-EOS AIRS Level 1A counts data (235 granules), our compression algorithm was empirically demonstrated to achieve a substantial compression ratio on the order of 3.7 to 1. Compression ratios achievable by our method for these hyperspectral data sets compare very favorably with what is currently achievable by other approaches as of the August 2005 special session of the SPIE Conference on Satellite Data Compression, Communications, and Archiving.

Plotted in Figure 16 is a normalized histogram of the compression ratios derived from the adaptive clustering compression algorithm applied to AIRS L1A global data from 8 November 2005. The mode of the distribution is 3.70 to 1, with the mean slightly higher (3.72) due to the positive skew evident in the results. A small number of cases had compression ratios less than 3.6, the minimum being 3.46. However, 90% of the cases had compression ratios between 3.61 and 3.80. The statistics of the sample are summarized in Table 1. Because of the non-Gaussian nature of the distribution, we make use of robust statistical measures instead of standard measures.



**Figure 16.** Statistical bootstrap histogram of mean compression ratios derived from random resampling of the data.

## 10. CONCLUSION

We have presented an exposition of the lossless compression method developed for the AIRS digital counts. We provided the motivation for using our approach, which combines the advantages of a clustering with linear modeling. We presented a number of visualizations which help clarify why our approach is particularly effective on this dataset. Our compression algorithm was empirically demonstrated to achieve a substantial compression ratio on the order of 3.7 to 1. The technique is continuing to undergo refinement and further development.

## 11. ACKNOWLEDGMENTS

This work is being sponsored by NOAA/NESDIS and has been prepared in support of the NOAA/NESDID satellite hyperspectral sounder data compression research group led by Roger Heymann of its Office of Systems Development and Tim Schmit of its Office of Research and Applications.

## REFERENCES

1. D. Salomon, *Data Compression: The Complete Reference*, Springer, New York, 2004 (third edition).
2. H. H. Aumann, M. T. Chahine, C. Gautier, M. D. Goldberg, E. Kalnay, L. M. McMillin, H. Revercomb, P. W. Rosenkranz, W. L. Smith, D. H. Staelin, L. L. Strow, and J. Susskind, "AIRS/AMSU/HSB on the Aqua Mission: design, science objectives, data products, and processing systems," *IEEE Trans. Geosci. Remote Sensing* **41**(2), pp. 253–264, 2003.
3. M. Goldberg, Y. Qu, L. M. McMillin, W. Wolf, L. Zhou, and M. Divakarla, "AIRS near-real-time products and algorithms in support of operational numerical weather prediction," *IEEE Transactions on Geoscience and Remote Sensing* **41**, pp. 379–389, 2003.
4. D. Huffman, "A method for the construction of minimum redundancy codes," *Proc. of IRE* **40**, pp. 1098–1101, 1952.
5. C. E. Shannon, "Communications in the presence of noise," *Proc. of IRE* **37**, pp. 10–21, 1949.
6. C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal* **27**, pp. 379–423, 623–656, 1948.
7. H. H. Aumann, D. T. Gregorich, S. L. Gaiser, D. F. Hagan, T. S. Pagano, L. Strow, and D. Ting, "AIRS level 1b algorithm theoretical basis document (ATBD) part 1 (IR)," (Technical Report), 10 November 2000.
8. G. Davis and A. Nosratinia, "Wavelet-based image coding: An overview," *Applied and Computational Control, Signals, and Circuits* **1**(1), 1998.
9. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification, S.E.*, Wiley Interscience, 2000.