Adaptive clustering for hyperspectral sounder data compression *

I. Gladkova†, L. Roytman†, M. Goldberg‡

† City College of New York, NOAA/CREST, 138th Street and Convent Avenue, New York, NY 10031
‡ NOAA/NESDIS, Office of Research & Applications, 1335 E/W Highway, Silver Spring, MD 20910

ABSTRACT

In this paper, which is part of an ongoing sequence of papers devoted to the subject of efficient noise-tolerant lossless compression of satellite data for transmission, we describe an algorithm for this purpose which effectively addresses the above criteria. Our algorithm exhibit the potential to achieve noise-tolerant compression ratios averaging 3.2 : 1. An earlier approach, which we presented at Third GOES-R User Conference[6] in May of 2004, was the first such method to break the 3 to 1 compression barrier for this class of data.

Keywords: compression, subspaces, clustering.

1. INTRODUCTION

The compression technique described in this paper arose from requirements imposed by several considerations which are desirable in the processing of data from NASA/NOAA's environmental satellites. The most important of these are speed, efficient memory utilization, and lossless compression. The motivation for a need for good compression algorithms is obvious. One example: data flow from the current NASA Aqua satellite can reach 89 GB/day.

Any algorithm which exhibits good performance in the above sense is a good candidate for on board processing of data prior to transmission. In addition to the above characteristics, such an algorithm should tolerate acceptably the inevitable presence of noise introduced during transmission.

With these criteria in mid, our algorithms are currently in the process of being tested as suitable interfaces with NOAA's existing data retrieval packages. When this testing is completed, we will report on the results.

This paper should be regarded as one of a sequence of papers, some in current development or preparation, devoted to the development and description of algorithms for the efficient transmission of satellite data.

A central principle of our work in this area is the attempt to cluster the data-points along or close to low-dimensional hyperplanes, since the actual satellite data we have been working with supports the hypothesis that such clustering in fact is present.

A key problem in such an approach is the identification of such hyperplanes, which turns out to be an interesting and challenging problem. A key requirement of any practical treatment of this problem is speed, because of current on board processing limitations.

2. DATA

Our group is working on compression techniques that are suitable for the next-generation NOAA/NESDIS Geostationary Operational Environmental Satellite (GOES) instruments. We are using current spacecraft to simulate data from the upcoming GOES-R instrument and focusing on Aqua Spacecraft's AIRS instrument in our case study.

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

Satellite Data Compression, Communications, and Archiving, edited by Bormin Huang, Roger W. Heymann, Charles C. Wang, Proc. of SPIE Vol. 5889 (SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.618085

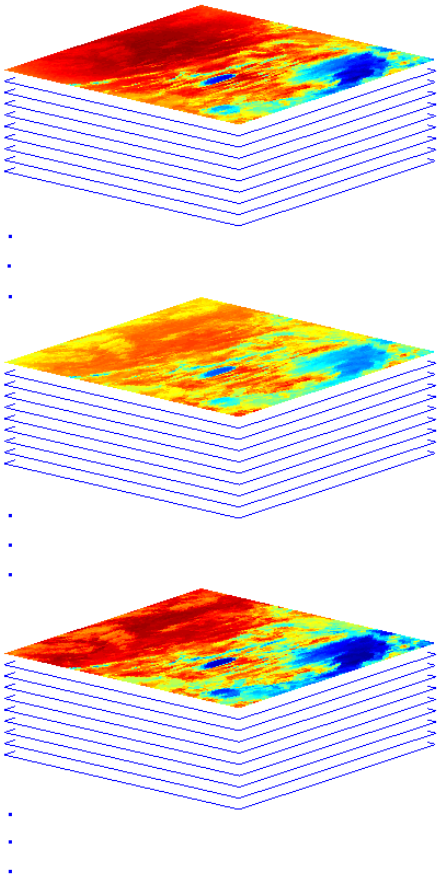

Figure 1. Hyperspectral image

The AIRS instrument is an infrared grating spectrometer with 2378 channels in the 3.74 micrometer to 15.4 micrometer spectral range. The AIRS takes 90 measurements as it scans 48.95 degrees perpendicular to the satellite's orbit every 2.667 seconds. We use Level 1A digital counts data granules, which represent 6 minutes (or 135 scans) of measurements. Therefore, our data set consists of a $90 \times 135 \times 2378$ paralelopiped of integers ranging from 12-14 bits.

AIRS data processing begins with Level 0 data from the spacecraft. The Level 1A product generation executives perform basic house keeping tasks such as data validation, ordering, geolocation refinement, and, finally, conversion of raw data numbers into engineering units. It is this Level 1A data that we are using in our case study.

More information about the data sets as well as instruments can be found on http://www-airs.jpl.nasa.gov/technology/technology_index.html. Figure 1 is a schematic of the above situation.

Note that noise in the channels introduce added complexity in compression. Therefore, in practice, we utilize only 1501 out of 2378 channels picked by NOAA for their favorable characteristics. Otherwise, we would need to add an additional step to detect these channels prior to running our compression algorithm.

3. DEFINITIONS AND NOTATIONS

Let an integer-valued function $r(x, y, z)$ be defined on integer lattice points $1 \leq x \leq N$, $1 \leq y \leq M$, $1 \leq z \leq K$, and assume that the range of r is $[0, L]$. For a fixed value of z, the function $\psi(x, y) = r(x, y, z)$ is an integer-valued N by M matrix. For use throughout the paper, we now introduce the term 'ψ-image' to refer to an arbitrary $N \times M$ array of reals, and note that for fixed z, the values $r(x, y, z)$ constitute a ψ-image. From this point of view, $r(x, y, z)$ can be regarded as a collection of ψ-images if we allow z to vary (see figure 1).

In the case of the hyperspectral data sets that will be considered in this paper, consecutive ψ-images of $r(x, y, z)$ are very similar and therefore contain somewhat redundant information. The purpose of this paper is to present a compression algorithm that uses this redundancy and allows us to represent the same information using less memory.

Let $\mathbf{R}^K$ be a Euclidian space of dimension K and D^K be a set of vectors $\mathbf{r} = (r_1, r_2, \ldots, r_K)$, where $r_j = r(x, y, j)$, and $1 \leq j \leq K$. The set D^K can clearly be regarded as a collection of vectors in $\mathbf{R}^K$, indexed by points (x, y). From this viewpoint, the task before us is to identify coherent clusters within these points of $\mathbf{R}^K$, in order to achieve efficient compression of the information they represent.

4. ALGORITHM

4.1. Normalization

The range $[0, L]$ of the function in question is known, since it depends on the measuring instrument's capabilities, therefore the set D^K is located inside a K-dimensional cube of volume L^K.

We will first normalize vectors of the set D^K to obtain a new set D on the unit sphere. We will retain the Euclidean norms $||\mathbf{r}||$ of vectors $\mathbf{r}$ of the original set D^K in order to restore the original data. This collection of norms will comprise the first of several ψ-images (see figure 2) that figure in our algorithm. We will denote this collection by $\psi_{||\mathbf{r}||}$. For the remainder of the paper, the notation $r(x, y, z)$ will refer to normalized $r(x, y, z)$.

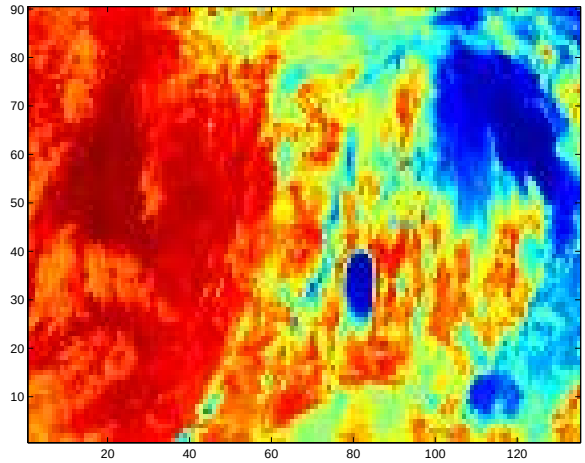

Figure 2. First ψ-image: $\psi_{||\mathbf{r}||}$, consisting of norms $||\mathbf{r}||$

4.2. Dimension

We start with an observation about the similarity between the consecutive ψ-images that form $r(x, y, z)$. As we progress from one image to the next, as illustrated by Figure 3, which displays the ψ-images for z equal to 653, 654 and 655 respectively, we see that they, like many other images in $r(x, y, z)$, are visually similar.

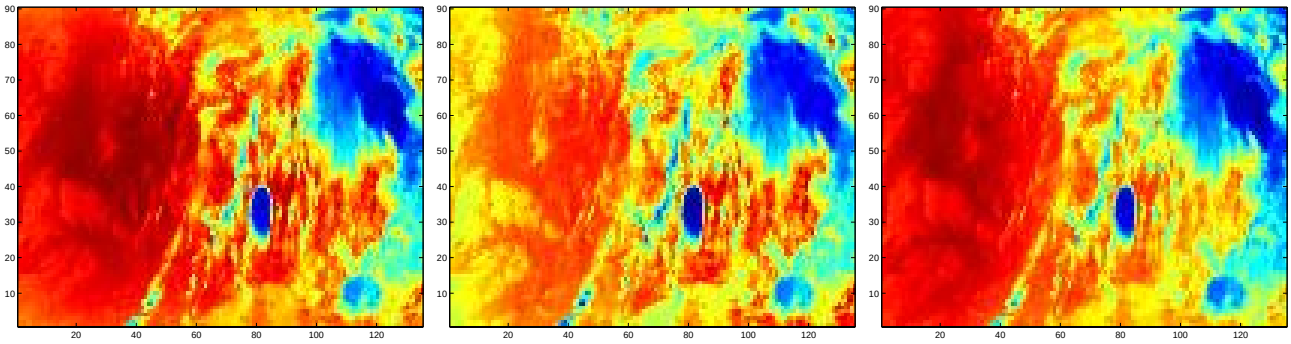

Figure 3. Three consecutive images from hyperspectral data set

Therefore the set D^K is stretched along the main diagonal of the cube. In the best case, one would find that the points are normally distributed about a line segment. In this case, projection on the principle directions will yield the best compression ratio. Most of the relations between consecutive images are distributed tightly about a line segment. This explains the extensive use of PCA (Principal Component Analysis) by NOAA scientists in data retrieval. Nevertheless, there are occasions (as illustrated in figure 4) when the distribution has more than one component.

Projection onto the eigenvectors of the covariance matrix is known to be optimal in the presence of a normal distribution of the data points. This is a well established technique for dimensionality reduction and a set of theorems about optimal properties of this transform can be found in numerous texts on multivariate analysis (see for example[9]).

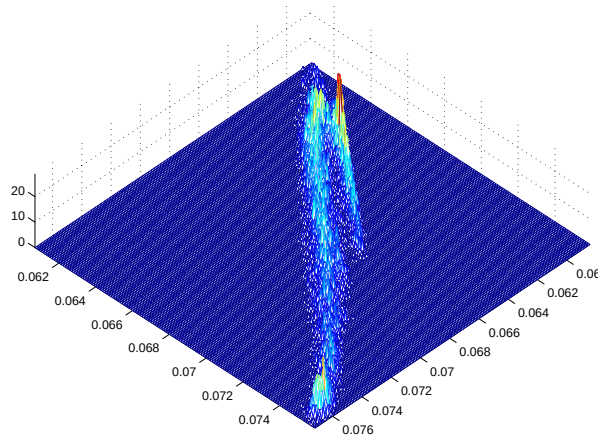

Figure 4 a

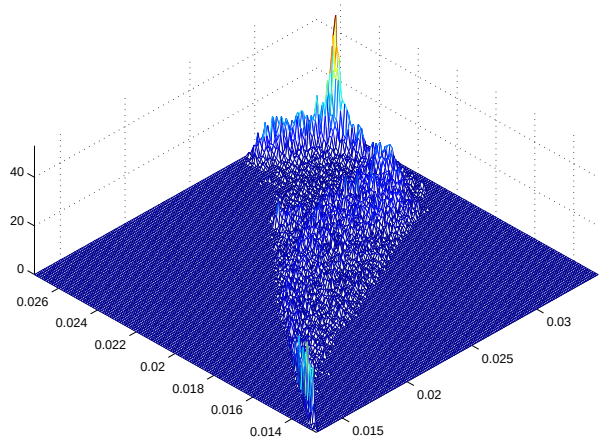

Figure 4 c

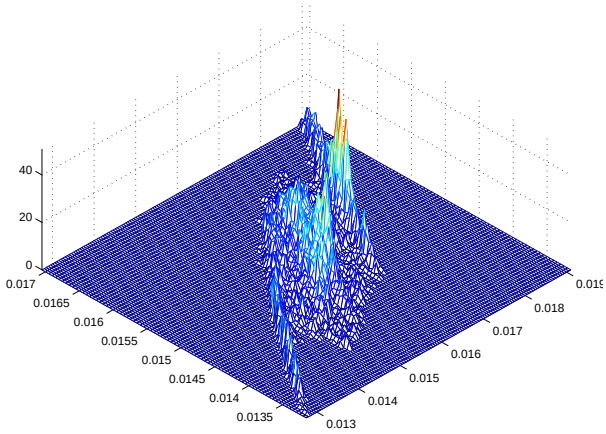

Figure 4 b

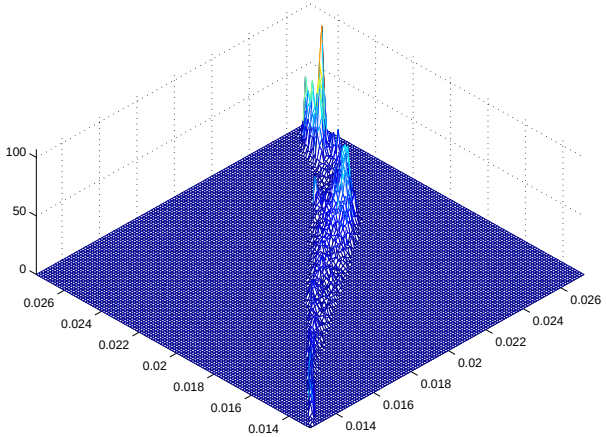

Figure 4 d

Figure 4. Distribution of points in set D

The data set in question does not follow the normal distribution (as figures 4a)-d) illustrates) and therefore the projection onto principal directions might not be optimal.

As was mentioned earlier, the set D is a K-dimensional cloud of points that is stretched along the diagonal, and numerous hyperspectral data sets have confirmed this characteristic of such data populations. The distribution of the points in D consists of multiple clusters approximately grouped in linear hyperplanes. For visual clarity, our illustrations (figure 4) are 2-dimensional projections of a K-dimensional configuration.

As can be noted from figures 4a)-d), different clusters have different mean and principal projection directions. Further improvements of the compression ratios can be achieved by subjecting the data to a suitable algorithm, such as the one described in this paper. That is, the set D should be projected onto several directions, rather then one.

There exists considerable literature on this and related subjects (cf. the references at the end of this paper, where a very short partial list is provided). In the following section we will describe a fast clustering algorithm for this particular type of data sets.

4.3. Adaptive Clustering

In this section, we illustrate the clustering component of our algorithm in the simplified case, chosen for visual clarity, of two consecutive ψ-images taken from a typical granule of 1501 images.

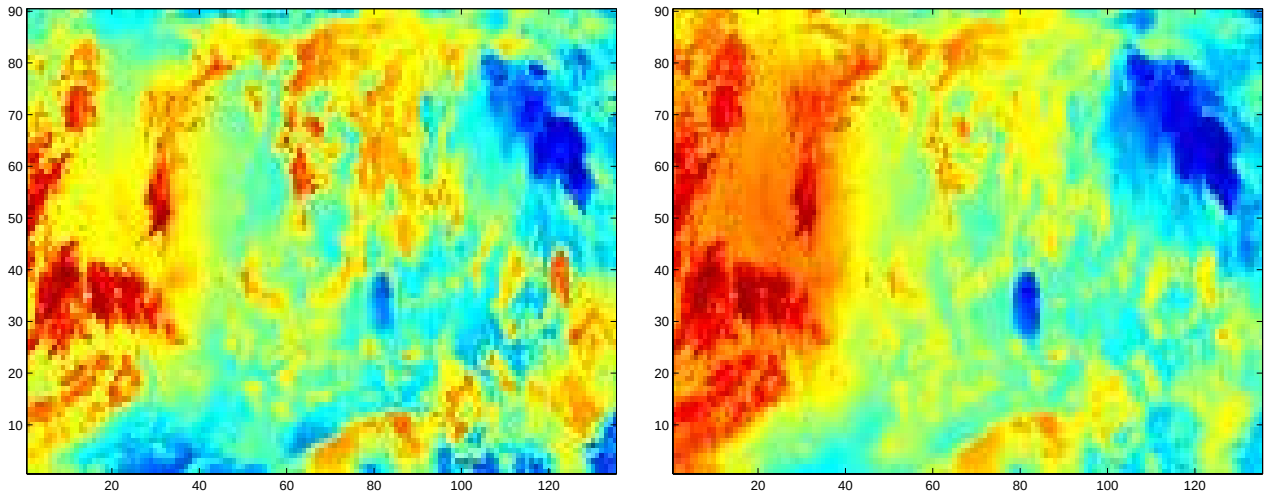

Figure 5. Example of normalized consecutive ψ-images corresponding to channel indexes 796 and 797

The collection of blue dots in figure 6 is a scatter plot of the gray level values of the pair of consecutive images shown in figure 5. Each dot represents a pixel in the image, with the horizontal coordinate being the gray value of the first image and the vertical coordinate being the gray level value of the second image.

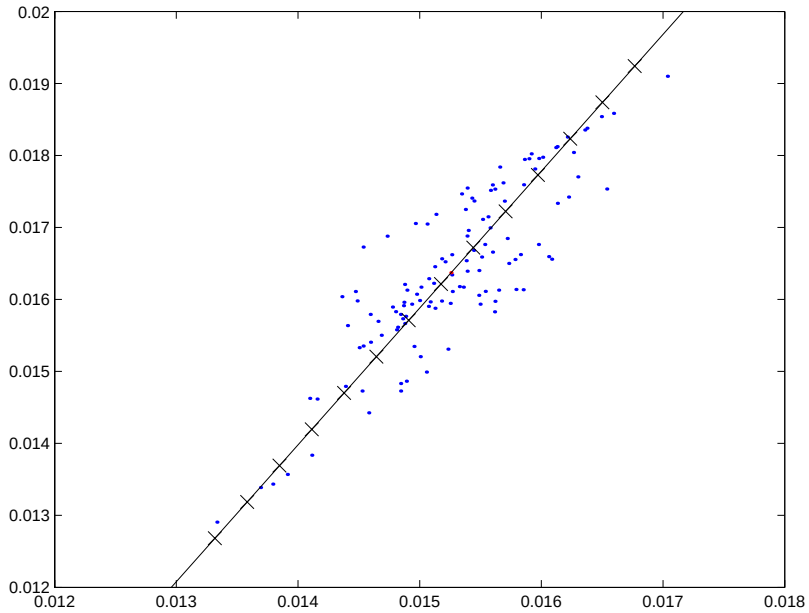

Figure 6. Scatter plot

The clustering procedure is as follows:

We begin by finding a least squares best fit line L for the entire initial set D of data points. We next equipartition L into equal-length segments, which results in a grouping of the points of D into 'bins,' distinguished by the segments of L onto which a point projects. Figure 6 illustrates this process. In figure 6, note the line L, the division points, and the corresponding intervals which define the bins. To avoid possible ambiguous assignments of points to a bin, we are taking the intervals as being semiopen.

Once the points of D are assigned to bins, we repeat the process, in the sense that we find the best least squares line fit to the points in each bin. Figure 7 illustrates this process, where we have indicated lines by segments, which have been drawn within the relevant bin. It is these segments with which we will next be working.

As this process of generating segments proceeds, we dynamically group some bins and ignore others, according to the following protocol:

If the points of origin and directions of two successive segments, as measured along L, are not within a prespecified proximity, we pass over the first of these segments, and examine the next pair, starting with the second of the previous pair. On the other hand, if two successive segments, as measured along L, have both their points of origin and their directions within the specified proximity, we coalesce their associated bins into a new bin but retain the last segment used, and then continue the above process, comparing the retained last segment with the segment corresponding to the next successor bin, until we encounter a bin whose segment does not match the preceding segment within the specified proximity. Should this occur, we do not combine that bin with the previously constructed bin. No further additions to the previous bin are permitted, and we re-initiate the above process, starting with the bin whose segment did not match.

There are obviously two extremes which could theoretically occur:

1. Everything is coalesced into one bin;

2. Everything remains divided into the original unaltered bins.

If event 1 occurs, we make no further attempt to achieve clustering, and proceed directly to encoding the data from D via a PCA method. If event 2 occurs, we go back and divide L more coarsely, and start over.

If we examine figure 7, this process would lead to a combining of the first three bins into one bin, the fourth segment could in principle start a new bin. However, it doesn't, since the fifth segment doesn't match it well.

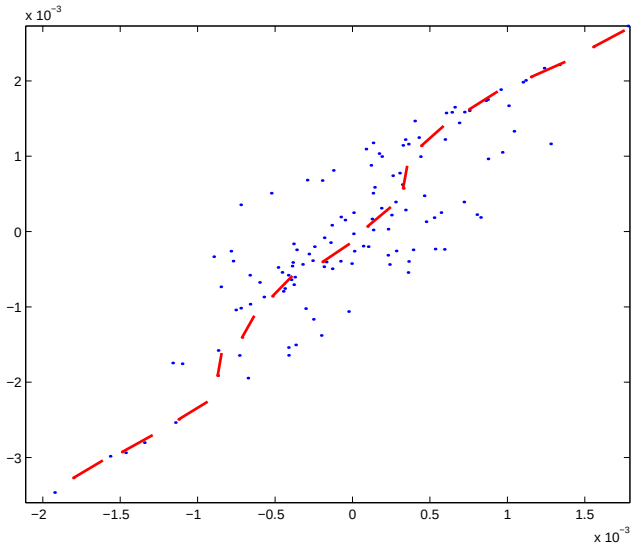

Figure 7

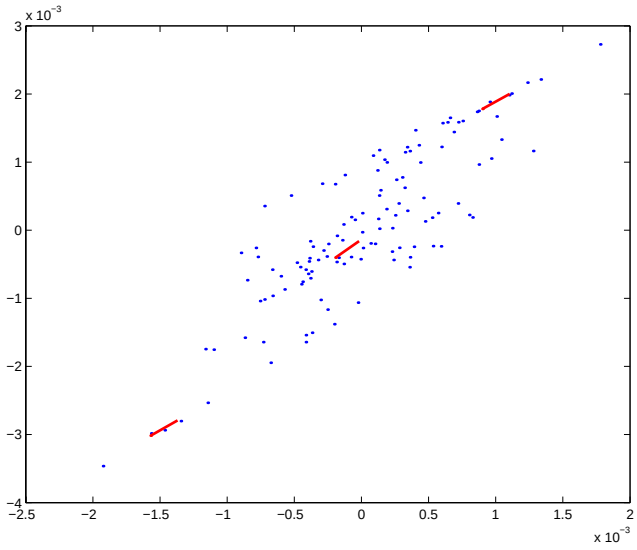

Figure 8

After a final division of D into bins has been achieved in the above way, we find a best least squares fit line for each resulting bin and then associate each point in D to one of these lines according to the criterion of shortest distance, with assignment in case of ties determined by order of processing. Figure 8 indicates the lines, which result from this process, and figure 9 indicates the corresponding assignment of points in D, indicated by color.

The final step consists of least squares recalculation of the just found lines $\ell \rightarrow \ell'$, which proceeds as follows: We consider the points D_j of D which are associated in the above way with a line ℓ_j. The set D_j may now of course include points which did not figure in the least squares computation which generated ℓ_j, and may similarly fail to include points from the bin whose points were used for the determination of ℓ_j. The result of this calculation is a possibly different line ℓ'_j.

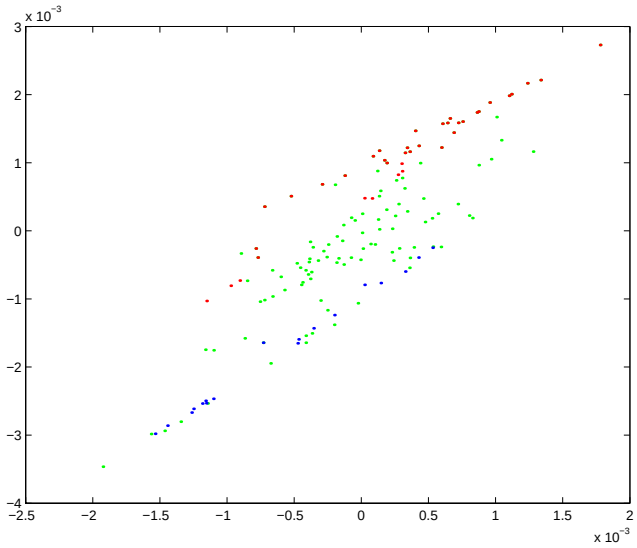

Figure 9

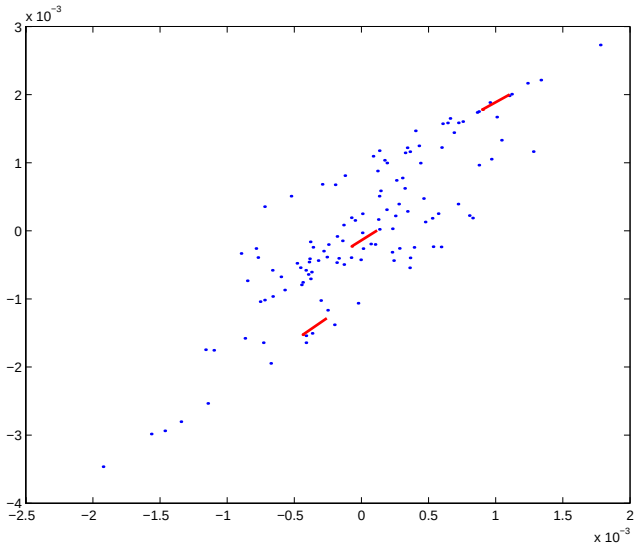

Figure 10

Figure 11 illustrates the clustering behavior which our algorithm has detected, when it is applied to the 2-image granule in figure 5. Each cluster is represented in a different color (red, green, blue), as was done in the case of figure 9. It is evident from figure 11 that our algorithm has detected three clusters in this case.

The 'cluster map' will be saved in order to correctly reassociate data points after decompression.

The compression is carried out separately for each cluster, with the appropriate cluster association retained as indicated above.

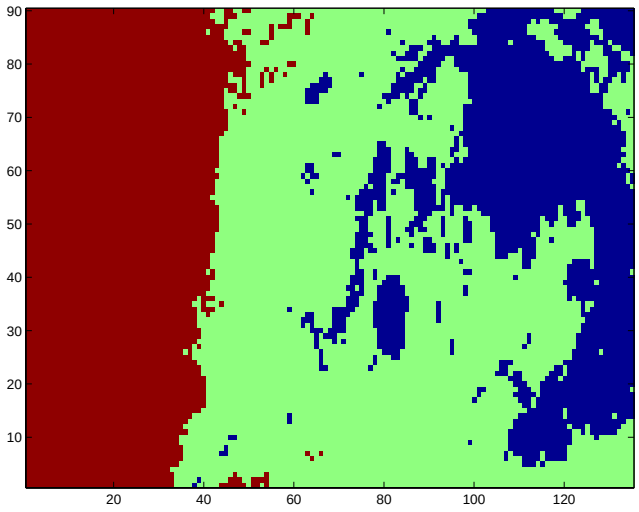

Figure 11

Each of the subsets D_j of data-points will be compressed independently via standard techniques: projection onto principal directions (PCA) and entropy encoding of orthocomplements of these projections.

4.4. Practical Considerations

For expository clarity, we have carried out our illustration of the algorithm in a 'toy' case consisting of only two slices. The actual AIRS data contains 1501 slices, so although the algorithm is in principle the same, in the interests of speed, we adapt our procedure for large dimensional applications in the following way:

We adjust the vectors $r \in D$ to each have mean (average of components) zero. We next select a longest vector , and then select a longest vector from its orthocomplement. We then select a longest vector from the orthocomplement of the two previously selected vectors, and continue in this fashion up to dimension fifty, by which point the lengths of the last vectors in the process become small. We then normalize the collection of forty vectors to facilitate projection on the subspace S they generate, and project the full set D onto S, calling the result D_S. The set D_S eventually became an input into our algorithm.

This procedure is of course a straightforward adaptation of the Gram-Schmidt process. The value of this approach is that it tends to select the most significant vectors out of data set which, empirically, seem to be largely characterized by relatively small numbers of vectors. For example, previous treatments of these kinds of data have focused on around 100 principle directions of the relevant covariance matrix.

4.5. Values to be stored

The values that are needed to reconstruct the original data set $r(x, y, z)$ will be stored in the compressed file. The are:

1) header containing relevant parameters,
2) cluster map indicating Q clusters,
3) $\psi_{||\mathbf{r}||}$ – collection of norms $||\mathbf{r}||$ rounded to integer values,
4) Q matrices $A = \{A_1, A_2, \ldots, A_{n_q}\}$ containing basis vectors A_z that define Q subspaces $\mathbf{R}^{n_q}(D_q)$
5) n_q ψ-images containing projection coefficients,
6) orthocomplements $\widetilde{r}(x, y, z)$.

We should remark that since all of the values in 3) trough 5) will be stored in the rounded form, we may not be able to reconstruct $r(x, y, z)$ without error that might occur due to the round off. Therefore, we will store differences $r'(x, y, z)$ between the original (not normalized) values $r(x, y, z)$ and their approximations computed from rounded values 2) – 5), rather than orthcomplements $\widetilde{r}(x, y, z)$.

Finally, we have collection of differences $\mathbf{r}'$ that are approximately normally distributed (see figure 12) and have variance at the level of instrument noise. For visual clarity, our illustration (as in figure 4) is a 2-dimensional projection of the total K-dimensional collection of differences $\mathbf{r}'$.

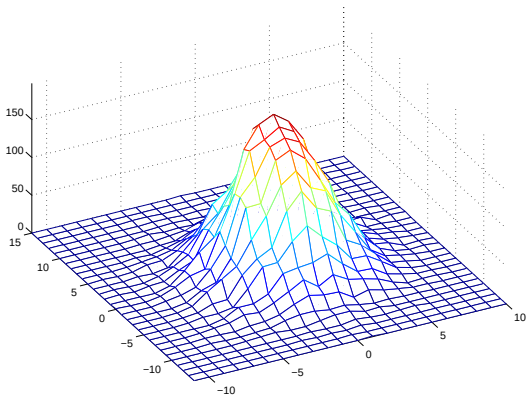

Figure 12. Distribution of differences $\mathbf{r}'$

The differences are compressed with entropy coding, and we build our Huffman codebook[8] based on a normal distribution, with variance computed from the differences $\mathbf{r}'$, so that the actual codebook does not have to be transmitted.

4.6. Treatment of transmission error

A consideration of paramount importance in practice is the integrity of the data transmitted via the algorithm in the presence of transmission noise.

To cope with this problem, we have developed simple and effective procedures, which preliminary testing indicates, handle the problem very well.

Briefly in outline, we have found that there is a simple hierarchy of criticality that can be easily associated with various subpopulations of the data to be transmitted.

For example, the data associated with the adaptive part of our algorithm is particularly sensitive, and we treat its storage separately. This inevitably somewhat lowers the achievable compression, but we have found a good balance between the conflicting goals of transmitted data integrity and compression.

Next, the data which figures in the PCA calculations is of lower criticality, but sufficiently vulnerable to effects of noise contamination to merit its own protection from noise effects, and we have developed a suitable less costly protection scheme for this class of data.

Lastly, the data that is entropy encoded is of the least criticality, but still requires some protection from noise effects, and we have developed a still less costly appropriate protection scheme for this class of data.

The detailed description of these procedures is the subject of a separate paper, currently in preparation.

It is of great importance to note that compression schemes which are not accompanied by meticulous attention to the effects of transmission noise are of little practical value in dealing with transmission over noisy channels.

5. RESULTS

In the following table, we give ratios of lossless compression for our 10 test granules (full 1501 channels). Note the slight penalty produced by our error-correction, which have integrated into our compression scheme.

We should note that these ratios are higher than those presented in our previous work[4] and that our methods, which were first announced at Third GOES-R User Conference[6] in May of 2004, where the first to break the 3 to 1 compression barrier for this class of data.

Granule	Location	Ratio (not error tolerant) (for noiseless channels)	Ratio (error tolerant) (wireless transmission)
9	Pacific Ocean, Daytime	3.5018	3.2630
16	Europe, Nighttime	3.5099	3.2415
60	Asia, Daytime	3.5009	3.1661
82	North America, Nighttime	3.5138	3.2736
120	Antarctica, Nighttime	3.5017	3.2560
126	Africa, Daytime	3.5005	3.1462
129	Arctic, Daytime	3.5168	3.2300
151	Australia, Nighttime	3.4957	3.1717
182	Asia, Nighttime	3.4980	3.1350
193	North America, Daytime	3.5004	3.1385

6. CONCLUSION

In this paper, which is part of ongoing series, we have presented a fast adaptive clustering algorithm for the compression of hyperspectral data sets. Our algorithm has achieved substantial compression, speed of execution, efficient memory utilization, and excellent noise tolerance.

7. 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. T. Boult, L. Brown, Factorization-based segmentation of motions, In *IEEE Workshop on Motion Undertsanding*, (1991) 179-186
2. Y. Cheng, Mean Shift, Mode Seeking, and Clustering, IEEE Transaction on Pattern Analysis and Machine Intelligence, 17(8) (1995) 790-799
3. D. Comaniciu, P. Meer, Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE Transaction on Pattern Analysis and Machine Intelligence, 24(5) (2002) 603-619
4. I. Gladkova, L.Roytman, M. Goldberg, J. Weber, Compression of AIRS data using Empirical Mode Decomposition, *Proc. of SPIE* 5548 (2004) 88-98
5. I. Gladkova, V. Popov, M. Goldberg, L.Roytman, Local least squares algorithm for satellite data compression (submitted for publication)
6. I. Gladkova, L. Roytman, M. Goldberg, J. Weber, Design of a Compression Algorithm for GOES Data, Third GOES-R User Conference, May 10-13, 2004, Boulder, CO
7. I. Gladkova, V. Popov, M. Goldberg, L.Roytman, A method for grouping data points into linear subspaces (submitted for publication)
8. D. Huffman, A method for the construction of minimum redundancy codes. *Proc. of IRE*, 40:1098–1101, 1952
9. S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1998
10. C.E. Shannon, Communications in the presence of noise, In *Proc. Of the IRE*, vol. 37, pp.10-21, Jan.1949
11. M. Tipping, C. Bishop, Mixtures of probabilistic principal component analyzers, Neural Computation, 11(2) (1999) 443-482
12. M. Tipping, C. Bishop, Probabilistic principal component analysis, Journal of the Royal Statistical Society, 61(3) (1999) 611-622
13. B. Schölkopf, A. Smola, K. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation, 10(5) (1998) 1299-1319
14. X. Zhuang, Y. Huang, K. Palaniappan, Y. Zhao, GaussianMixture Density Modeling, Decomposition, and Applications, IEEE Transactions on Image Processing, 5(9) (1996) 1293-1302