# Priority-Based Error Correction Using Turbo Codes for Compressed AIRS Data *

**I. Gladkova,**[1] **M. Grossberg,**[1]
**E. Grayver,**[2] **D. Olsen,**[2]
**N. Nalli,**[3] **W. Wolf,**[3] **L. Zhou,**[3]
**M. Goldberg**[4]

[1] CCNY, NOAA/CREST, 138th Street and Convent Avenue, New York, NY 10031
[2] The Aerospace Corporation, El Segundo, CA 90245
[3] QSS Group, Inc. NOAA/NESDIS, E/RA1 5211 Auth Road, Camp Springs, MD 20746
[4] ORA NOAA/NESDIS, E/RA1 5200 Auth Road, Camp Springs, MD 20746

## ABSTRACT

Errors due to wireless transmission can have an arbitrarily large impact on a compressed file. A single bit error appearing in the compressed file can propagate during a decompression procedure and destroy the entire granule. Such a loss is unacceptable since this data is critical for a range of applications, including weather prediction and emergency response planning. The impact of a bit error in the compressed granule is very sensitive to the error's location in the file. There is a natural hierarchy of compressed data in terms of impact on the final retrieval products. For the considered compression scheme, errors in some parts of the data yield no noticeable degradation in the final products.

We formulate a priority scheme for the compressed data and present an error correction approach based on minimizing impact on the retrieval products. Forward error correction codes (e.g., turbo, LDPC) allow the tradeoff between error correction strength and file inflation (bandwidth expansion). We propose segmenting the compressed data based on its priority and applying different-strength FEC codes to different segments. In this paper we demonstrate that this approach can achieve negligible product degradation while maintaining an overall 3-to-1 compression ratio on the final file. We apply this to AIRS sounder data to demonstrate viability for the sounder on the next-generation GOES-R platform.

**Keywords:** compression, hyperspectral data, spectral clustering, multiple eigenspaces, channel coding, error propagation

## 1. INTRODUCTION

Global measurements from high spectral resolution infrared (IR) sounders will consist of a large volume of global geophysical data in near-real time. Such data will be obtained from several IR instruments to fly onboard satellites in both sun- and geosynchronous orbital configurations, including the AQUA-EOS Atmospheric Infrared Sounder (AIRS) (launched May 2002), the NPOESS Cross-track Infrared Sounder (CrIS), the MetOp Infrared Atmospheric Sounding Interferometer (IASI) and the GOES-R Hyperspectral Environmental Sounder (GOES-R HES). While the purpose of gathering this data is to improve geophysical parameter retrievals and model assimilation, the data poses a serious challenge in data management, including archival and distribution to the science community. Thus, to prepare for this high spectral resolution data, compression techniques are necessary and are currently under research and development (e.g.,[1–6]). Such techniques must not only achieve an adequate compression ratio, but they must also be robust to transmission errors that can occur during the satellite downlink to ground stations.

In this work we test a new lossless compression technique that was presented by Gladkova et al.[7] for error containment. This compression technique was developed to specifically address the key requirements that are very important in the processing of data from the National Oceanic and Atmospheric Administration (NOAA)/NASA's environmental satellites. The most important of these are high lossless compression ratios and error robustness. Our algorithm exhibits good performance in the above scenarios. Thus, it is a good candidate for handling transmission and distribution through wireless communication channels. It is very important that the compression/decompression algorithms should acceptably tolerate the inevitable presence of noise introduced during transmission.

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

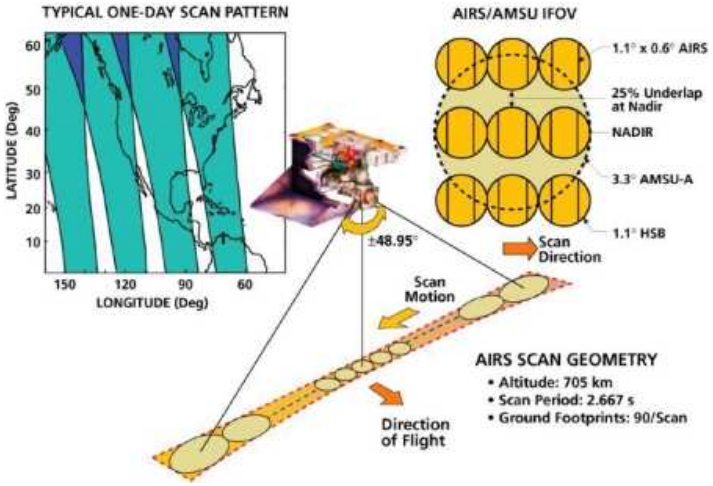


**Figure 1.** AIRS scan geometry.

We make use of the fact that under the considered lossless compression scheme, the data is stratified into two components. One component is the projection of the data into a space that can be parameterized by the model. The second component is the residual error. In this paper, we propose to use variable-strength error correction to take into account the fact that the sensitivity of the reconstruction error is concentrated in the component of the compressed data that has been fit to the model. This makes it possible to minimize impacts of errors on reconstruction error without significantly increasing file size.

The compression ratio of the algorithm has already been established empirically to be 3.7 to 1 using AIRS Level 1A (L1A) data. Here, for the first time, we employ an end-to-end system that uses AIRS L1A data and undergoes lossless compression, simulated transmission through a noisy channel by means of satellite forward error correcting coding, decompression, and processing though the NOAA/NESDIS (National Environmental Satellite, Data, and Information Service) near-real-time AIRS processing system, which outputs temperature, water vapor, ozone, and trace gas atmospheric retrievals. The retrievals were then compared with the retrievals produced using the original AIRS counts. The results of this end-to-end test did not indicate any negative impact on sounder products.

## 2. DATA

AIRS is an IR grating spectrometer that measures Earth radiance between 650 $cm^{-1}$ and 2675 $cm^{-1}$ in 2378 channels.[8] This data is processed operationally at NOAA/NESDIS. The near-real-time AIRS data provides the opportunity to use empirical satellite observations as a test bed for data compression studies of future hyperspectral sounding instruments.

In the current demonstration, we use 2 AIRS granules, one from 8 November 2005, the other from 10 July 2006. Each of these AIRS data granules consists of IR spectra for 90 by 135 fields of view (FOV). The 2 granule files were produced from the NOAA/NESDIS Near Real-Time (NRT) AIRS Processing System. This system has been operational since October 2002, distributing AIRS data to the Numerical Weather Prediction Centers within three hours of acquisition. The compression algorithm operates on Level 1A navigated data stored as counts. This data, stored originally in HDF-EOS files, is broken down into a channel subset of 1501 channels. This channel reduction forms a dataset of stable channels that is representative of both grating spectrometer data (e.g., AIRS) and Michaelson interferometer data (e.g., CrIS, IASI, GOES-R). This data provides the perfect opportunity to evaluate compression and error correction algorithms in preparation for the future hyperspectral satellite data.

## 3. SYSTEM DESCRIPTION

Results presented in this paper are based on a combination of emulation and simulation. A high-fidelity emulation of the satellite downlink is provided by the flexible GOES-R test bed. The test bed[9] consists of a programmable digital modem, radio frequency subsection, a high-power space-qualified traveling wave tube amplifier, and a calibrated noise source.

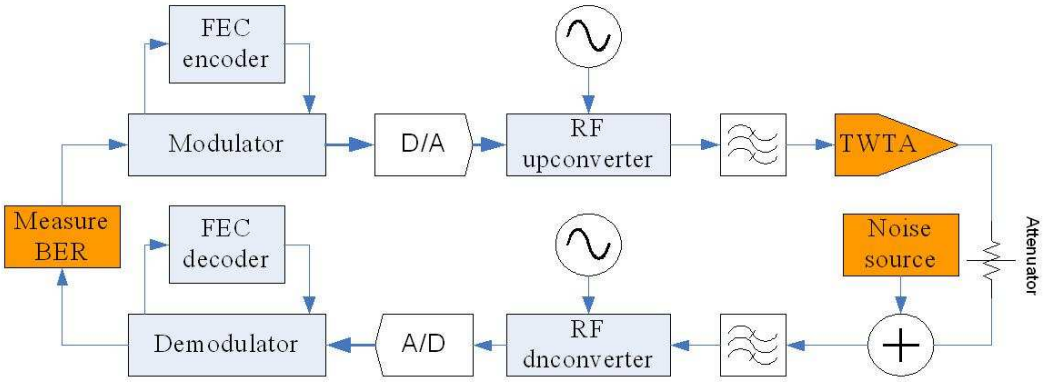


**Figure 2.** GOES-R test bed block diagram.

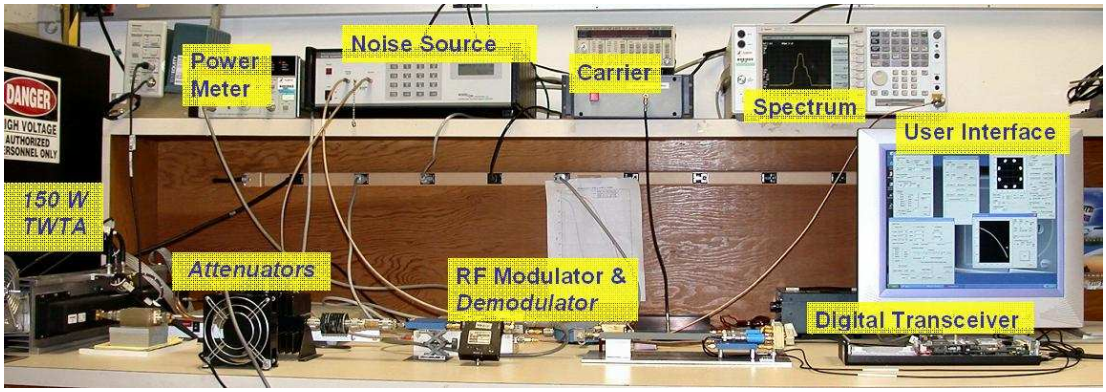


**Figure 3.** Aerospace GOES-R test bed photo.

The emulation results include most of the significant sources of error and distortion – nonlinear distortion, thermal noise, implementation loss in the RF, mixed signal, and digital components. The implementation loss in the modem accounts for carrier, timing, and amplitude tracking loops.

The test bed was used to collect bit error patterns for different combinations of signal-to-noise ratio ($E_b/N_0$) and error correction coding. The downlink channel is broadcast and unidirectional, with no re-transmit capability. Three target bit error rates (BER) were considered: $10^{-7}$, $10^{-8}$, $10^{-10}$. Hundreds of errors were detected for each scenario, and the location of each bit error in a file (granule) was recorded. The real-time, hardware-based collection of errors ensures accurate measurement of the correlation between error locations. The errors after FEC decoding tend to come in clusters rather than being uniformly distributed. Thus, thousands of error-free blocks may occur, followed by a block with multiple errors. The bit error positions were then used as a mask to flip the corresponding bits in the compressed data. The overall link quality is judged based on the science output from the decompressed data.

## 4. OVERVIEW OF COMPRESSION APPROACH

### 4.1. 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.[10,11] We have chosen to work with digital counts rather than post-processed rounded versions 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 $K = 1501$ infrared (IR) spectral channels for each footprint. There are $I = 90$ footprints per scan line and $J = 135$ scanlines are grouped to make up a granule.

These $I \times J \times K$ hyperspectral data cubes correspond to a bounded integer-valued function $q(x, y, \nu)$ defined on the domain 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 in a geospatial grid, and the $\nu$-corresponds to the sensor's spectral channel. The values $q$, or digital counts, are quantizations of an analog signal that 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[12] but is not crucial here, since our compression method operates directly on digital counts. The quantization into digital counts has a range of $[0, 2^{N_{\text{bits}}} - 1]$ and $N_{\text{bits}}$ 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, $N_{\text{bits}} = 14$.

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.

### 4.2. Dimensionality Reduction

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 $Q$ tend to aggregate near considerably lower-dimensional linear subspaces 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.

If the data lives close to $N$ dimensional subspace of $\mathbb{R}^K$, the smallest $K - N$ eigenvalues will be very small. Hence, the data can be accurately approximated by projection into the space $V_N$ passing through the mean $\mu$ and spanned by the first $N$ eigenvectors vectors

$$\mathbf{a}_1, \quad \mathbf{a}_2, \quad \ldots, \quad \mathbf{a}_N. \tag{1}$$

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

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

The error vector

$$\widetilde{\mathbf{q}_\varepsilon} = \mathbf{q} - \mu - \rho_1 \mathbf{a}_1 - \ldots - \rho_N \mathbf{a}_N, \tag{3}$$

is a vector perpendicular to the subspace $V_N$.

### 4.3. Clustering

While a PCA-based linear model is capable of capturing correlations between different dimensions, it assumes that the data is clustered about a single hyperspace. If we have $M$ clusters, we can associate a matrix $A_m$ for 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

$$L(A_0, A_1, \ldots, A_M) = \sum_{\mathbf{q} \in \mathbf{Q}} \min_{0 \leq m \leq M} \mathbf{d}(A_m, \mathbf{q}). \tag{4}$$

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

The distance from a point $\mathbf{q}$ to the closest subspace spanned by the columns of $A_0$, $A_1$, ..., $A_M$ is defined by the minimum of $\mathbf{d}(A_m, \mathbf{q})$ with respect to the index $m$, and the Lagrangian in (4) is the sum of the squares of distances from every point $\mathbf{q}$ in $\mathbf{Q}$ to the closest subspace 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 principal subspace, it is closest to the cluster mean, thus resolving ties. So let $\mathbf{c}(\mathbf{q})$ be an assignment map that describes the above-determined assignments for each point $\mathbf{q}$.

The optimization of the objective function $L$ in (4) 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 $M$ subspaces $\mathbb{R}^N(A_m)$ defined by $\mathbf{a}_n$, $n = 1, \ldots, N$. 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 next iteration of the algorithm begins.

We should note that prior knowledge of clustering assignments is needed. The design of the algorithm provides this for every iteration except the first. We begin by finding a least squares best fit line $l$ for the entire initial set $\mathbf{Q}$ of data points. We next partition $l$ into segments with an approximately equal amount of data points, which results in a grouping of the points of $\mathbf{Q}$ into bins, distinguished by the segments of $l$ onto which a point projects.

The coefficients $\{\rho_n\}_{n=1}^N$ of projections onto corresponding subspaces $\mathbb{R}^N(A_m)$ are indexed by $(x, y)$. These can thus be interpreted as $N$ 2D images. These coefficients, the vectors $\mathbf{a}_n$, assignment map, and differences $\widetilde{\mathbf{q}}$ together make it possible to reconstruct the original file.

## 5. PRIORITIZATION SCHEME

There is a simple hierarchy of criticality that can be easily associated with various subpopulations of the data. We will discuss the hierarchy in this section and present the channel encoding procedures in the next section.

The data associated with the clustering part of our algorithm is particularly sensitive; any single bit flip in the part of the compressed file that contains the model parameters will result in severe error degradation of the reconstructed hyperspectral image. Fortunately, it occupies a very small amount of memory and can be easily protected without affecting the compression ratio. We have two layers of protection – error correction within the compressed file itself (extra layer) and by stronger channel coding (cf. next section for description of channel encoding).

Next, the data that figures in the local coordinates calculations is of lower criticality than model parameters, but sufficiently vulnerable to effects of noise contamination to merit its own protection from noise effects. It is protected in our scheme by the channel coding of the same strength as the parameters of the model, but without the extra layer.

Lastly, the data that is entropy encoded is of the least criticality, but still requires some protection from noise effects. It is protected in our scheme by a weaker channel encoding.

## 6. CHANNEL CODING

Forward error correction (FEC) encoding is the addition of redundancy, e.g., parity-check symbols, to a message that is to be transmitted over a medium (the communication channel). This redundancy allows the error correction decoder to detect and/or correct erroneous data and restore the received data stream to the original data stream. A variety of codes have been developed over the last five decades, starting with the simple parity check, progressing to a hard-decision Reed-Solomon (RS) code, and then to a soft-decision convolutional code. Convolutional codes have been a standard FEC technique until the recent development of turbo (e.g., TPC) and low-density-parity-check (LDPC) codes. An FEC code is typically defined by the following set of parameters: algorithm, block size, code rate. The code performance is characterized by a bit-error-rate (BER) curve. The BER curve plots the average number of errors against the signal-to-noise ratio ($E_b/N_0$). In general (but with many exceptions), the number of errors at a given $E_b/N_0$ decreases when block size increases or code rate decreases. The latter statement is intuitively obvious since lower code rates imply more redundant symbols that can be used to detect and correct errors. Most FEC algorithms can support a wide range of block sizes and/or code rates, thereby allowing a continuum of BER at a given $E_b/N_0$ by trading BER against throughput.

Choosing different FEC codes for different parts of the data stream is the key idea presented in this paper. As discussed in section 5, an error in a subset of the transmitted data causes many more errors in the final output data than an error

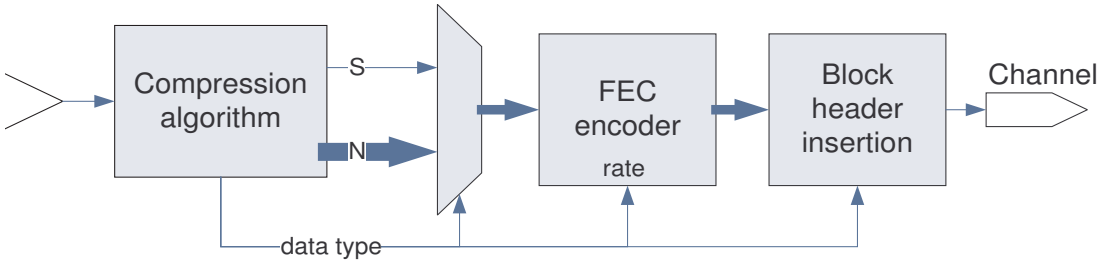


**Figure 4.** Block diagram of the dual-rate FEC encoder.

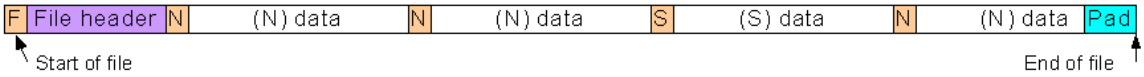


**Figure 5.** Notional framing structure.

in a different subset of the transmitted data. The more 'sensitive' part of the data is encoded with a stronger FEC code than the less sensitive part. Of course, the stronger code implies higher overhead and therefore lower overall compression. However, if the sensitive data is a small fraction of the total data, the effect is negligible. A block diagram of the proposed scheme is shown in Figure 4. Note that this scheme separates all data into only two types: sensitive (**S**) and normal (**N**). Some data streams may be separated into more types, with a different code assigned to each (see, for example,[13]).

The compressed data of each type is segmented into packets equal to the FEC block size for that data type. Each encoded block is then prefixed with a header. The header is used to indicate the data type and is required for the receiver to select the appropriate rate in the FEC decoder. The header may be as short as 1 bit. However, there is a small but nontrivial probability that the packet header will be received with errors. The probability of error is increased since it is not encoded. This problem can be addressed by simply repeating the header multiple times such that the probability of error in the repeated sequence is made negligible. Alternatively, the header for each packet may be added to the data of the previous packet. This approach results in somewhat longer latency and buffering requirements. The total number of compressed bits of either type may not be an integer multiple of the FEC block size. Thus, the last packet in the compressed data file for each data type may have to be padded with junk bits. The number of these bits may be part of the packet header, or communicated via a side channel. In either case, since the compressed data file consists of thousands of FEC packets, the overhead due to padding is negligible. A separate packet (**F**) with high-level information about the file contents is sent at the beginning of each file and encoded with the stronger code. A notional frame structure is shown in Figure 5.

## 7. CODE SELECTION

The discussion above motivates the need for a pair of strong and stronger FEC codes. This need can be fulfilled in a number of different ways, including:

1. Use one strong code and combine it with another code to get a stronger code. A typical example would be combining convolutional and Reed-Solomon codes.
2. Use codes at the same code rate but with different algorithms (e.g., encode (**N**) data with a Turbo code and encode (**S**) data with a convolutional code).
3. Use two different code rates with the same algorithm. This approach is preferred since it provides the highest throughput and reasonable hardware complexity.

Most FEC codecs available today provide run-time configurability and can change the code rate and block size on a block-by-block basis.[13,14] Two algorithms are considered for the system described in this paper: TPC and LDPC. These modern codes provide better performance than the legacy convolutional codes, and advances in semiconductor technology have made their implementation relatively low cost.

A basic TPC encoder processes an input block of $k_1 \times k_2$ and generates an output block of $n_1 \times n_2$, for a code rate $r = (k_1 k_2)/(n_1 n_2)$. The particular TPC codec used for this study (and most other readily available TPC codecs) restricts the values of $n \in \{128, 64, 32, 16, 8, 4\}$ and $k \in \{127, 120, 63, 57, 31, 26, 15, 11, 7, 4, 3\} \pm 1$. Note that lower code rates correspond to smaller block sizes. Codes with smaller block sizes typically exhibit an error floor higher than codes with larger block sizes. Thus, these lower-rate codes actually perform worse than higher-rate codes for very low BER. Furthermore, the implementation complexity increases slightly if the block sizes for different data types are not equal. Thus, two TPC codes with the maximum supported block size but different code rates are selected for (**N**) and (**S**) data. The two codes are:

$r = 0.878 : (128, 120) \times (128, 120)$ for (**S**) data

$r = 0.923 : (128, 120) \times (128, 126\text{e})$ for (**N**) data

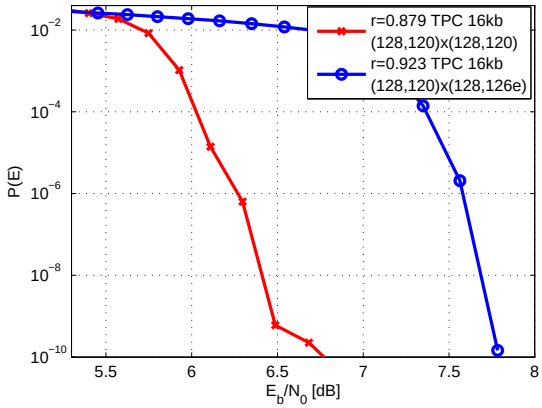


**Figure 6.** BER performance for 8-PSK for two TPC codes.

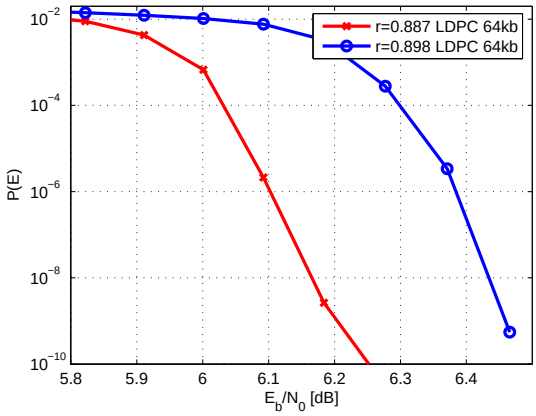


**Figure 7.** BER performance for 8-PSK for two LDPC codes.

LDPC coding is becoming a popular alternative to TPC. LDPC offers essentially identical performance and has comparable computational complexity. LDPC has been selected for error correction in the new digital satellite broadcast TV standard – DVB-S2. The standard specifies a block size of up to 64 kb and code rates from 0.5 to 0.9. The expected widespread use of this standard is likely to result in easily obtainable and low-cost decoders. Performance for two LDPC codes is given in Figure 7.

Either the TPC or LDPC code family can be used for the unequal protection coding scheme described above. The LDPC code has two advantages that motivate its selection as the new baseline code:

1. The decoders are likely to be widely available and low cost due to the high-volume application in the DVB-S2 receivers.
2. The DVB-S2 standard includes specifications for changing the code rate on a frame-by-frame basis.

The BER for both codes decreases rapidly for BER $< 10^{-8}$. Thus, if the target BER for the weaker code is $10^{-8}$, the expected BER for the stronger code is essentially 0. This observation motivates the simulations presented in section 8. The sensitive data is assumed to be error free, while the normal data has the target BER.

## 8. RESULTS

To evaluate the effectiveness of the error correction scheme, the AIRS corrected counts were processed to produce atmospheric retrievals and then compared to the retrievals produced from the original AIRS counts. To accomplish this processing step, a system was set up to evaluate errors in the resulting scientific products.

The transmitted, uncorrected AIRS data was written back into the AIRS Level 1A HDF-EOS format. This file was then processed though the NOAA/NESDIS/STAR near-real-time AIRS processing system[15] to produce calibrated and navigated

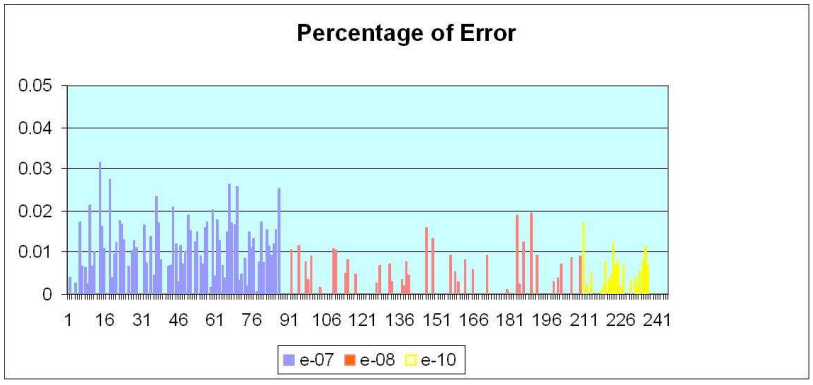


**Figure 8.** Percentage of counts that have different values than the original. Three colors represent different BER levels.

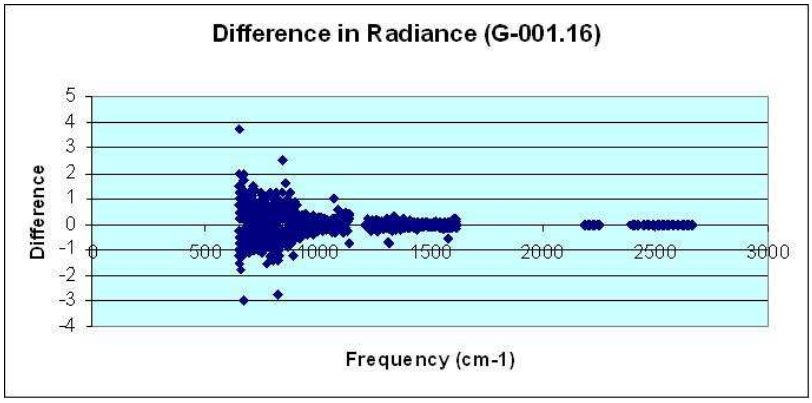


**Figure 9.** Differences between the original and error-corrected radiance.

AIRS radiances (using version 4.0.9.0 of the AIRS processing). These AIRS Level 1B radiances were then processed with the NOAA/NESDIS/STAR offline AIRS Level 2 processing system. The output of this processing step is temperature, water vapor, ozone, and trace gas atmospheric retrievals. The retrievals produced using the error corrected AIRS counts were then compared with the retrievals produced using the original AIRS counts.

Figure 8 displays the percentages of the cases that have different values between the error-corrected counts and the original AIRS counts. They are less than 0.035% for all 3 different BER levels. An important fact is that all of the 1501 channels processed are used in the regression step, which provides a first guess for the physical retrieval. Therefore any changes in the counts of this channel set could potentially affect the retrieval results. Figure 9 displays a typical spectrum of the differences between the original and error-corrected AIRS radiances.

Figure 10 shows the AIRS observed brightness temperature for granule-45 of November 08, 2005. It is a combination of land, ocean, coastal area, cirrus clouds, and possibly some dust contamination. This is a very difficult granule for the retrieval. To estimate the impacts, we have run retrievals for all cases with 3 different BER levels for this granule. Figure 12 displays the root mean square (RMS) errors of the retrievals of temperature (left), moisture (middle), and ozone (right), compared with the AVN (aviation global model) forecast model. The statistics reflect only the cases that passed the quality checks (37.9% of the total cases). As we can see, the profiles of RMS errors for different cases are almost exactly on top of each other. Table 1 shows the BIAS and RMS difference between the error-corrected cases and the original cases. Basically the BIASs are all zeros and the RMS errors are very small. The RMS errors are generally less than 1% of the required accuracies. We conclude that the difference resulting from the error-correction does not have a significant impact on the retrievals.

Table 1 summarizes the BIAS and RMS errors of temperature, moisture, and ozone retrievals for different pressure layers, from different BER levels. Temperature is given in degrees (K), moisture in $g/\text{cm}^2$, and ozone in DU (Dobson Unit).

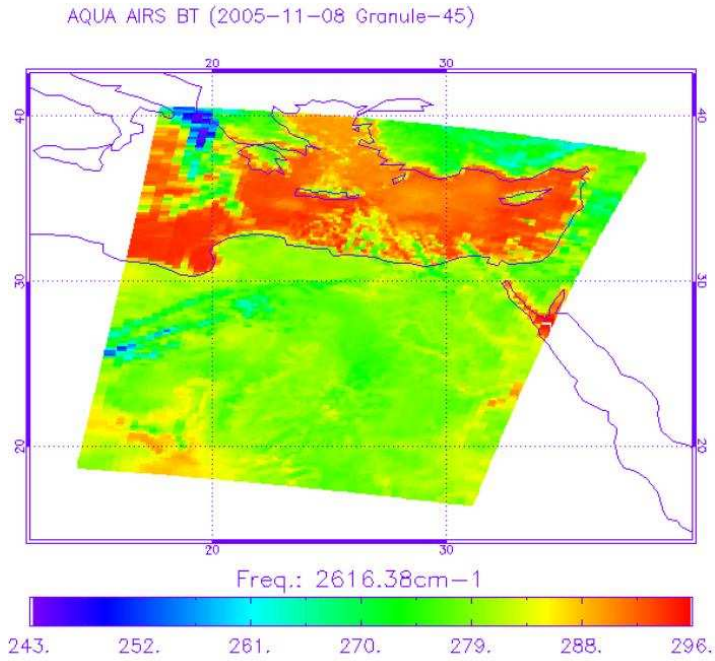


**Figure 10.** AIRS observed brightness temperature for channel $2616\text{cm}^{-1}$, over East China Sea, on Nov. 08, 2005.

| | | T (103mb-478mb) | T (478mb-778mb) | T (778mb-1100mb) | Q (201mb-314mb) | Q (314mb-618mb) | Q (618mb-1100mb) | O3 (total) |
|---|---|---|---|---|---|---|---|---|
| E-07 | BIAS | 0.00 | 0.00 | 0.00 | 0.0000 | 0.0000 | 0.000 | 0.0 |
| | RMS | 0.01 | 0.01 | 0.02 | 0.0000 | 0.0002 | 0.001 | 0.2 |
| E-08 | BIAS | 0.00 | 0.00 | 0.00 | 0.0000 | 0.0000 | 0.000 | 0.0 |
| | RMS | 0.01 | 0.01 | 0.01 | 0.0000 | 0.0000 | 0.001 | 0.1 |
| E-10 | BIAS | 0.00 | 0.00 | 0.00 | 0.0000 | 0.0000 | 0.000 | 0.0 |
| | RMS | 0.01 | 0.01 | 0.01 | 0.0000 | 0.0001 | 0.001 | 0.1 |

**Table 1.** BIAS and RMS errors of temperature, moisture, and ozone retrievals for different pressure layers, from different BER levels. T represents temperature, Q represents moisture, and O represents ozone. Temperature in degrees (K), moisture in $g/\text{cm}^2$, and ozone in Dobson Units (DU).

## 9. CONCLUSION

This paper is part of an ongoing series devoted to lossless compression of hyperspectral data. Our group is working on robust compression techniques that are suitable for the next-generation NOAA/NESDIS Environmental Satellite instruments. The research and development of our ongoing project specifically focuses on requirements that are very important in the processing of data from NOAA/NASA's environmental satellites, namely high lossless compression ratios and error robustness.

To address the first requirement, we have designed a lossless compression algorithm that exploits properties of hyperspectral imaging that are known a priori. This data-driven approach allowed us to achieve superior results over generic compression methods. The second requirement is imposed by the need to transmit the compressed data over band-limited noisy channels. It is crucial that compression/decompression algorithms acceptably tolerate the inevitable presence of noise introduced during transmission. We have developed an error-robust approach that is based on minimizing the impact on the retrieval products. We believe that the best measurement of performance, i.e., how the algorithm copes with error propagation issues, is the impact on the actual products that are derived from raw instrument measurements. To perform

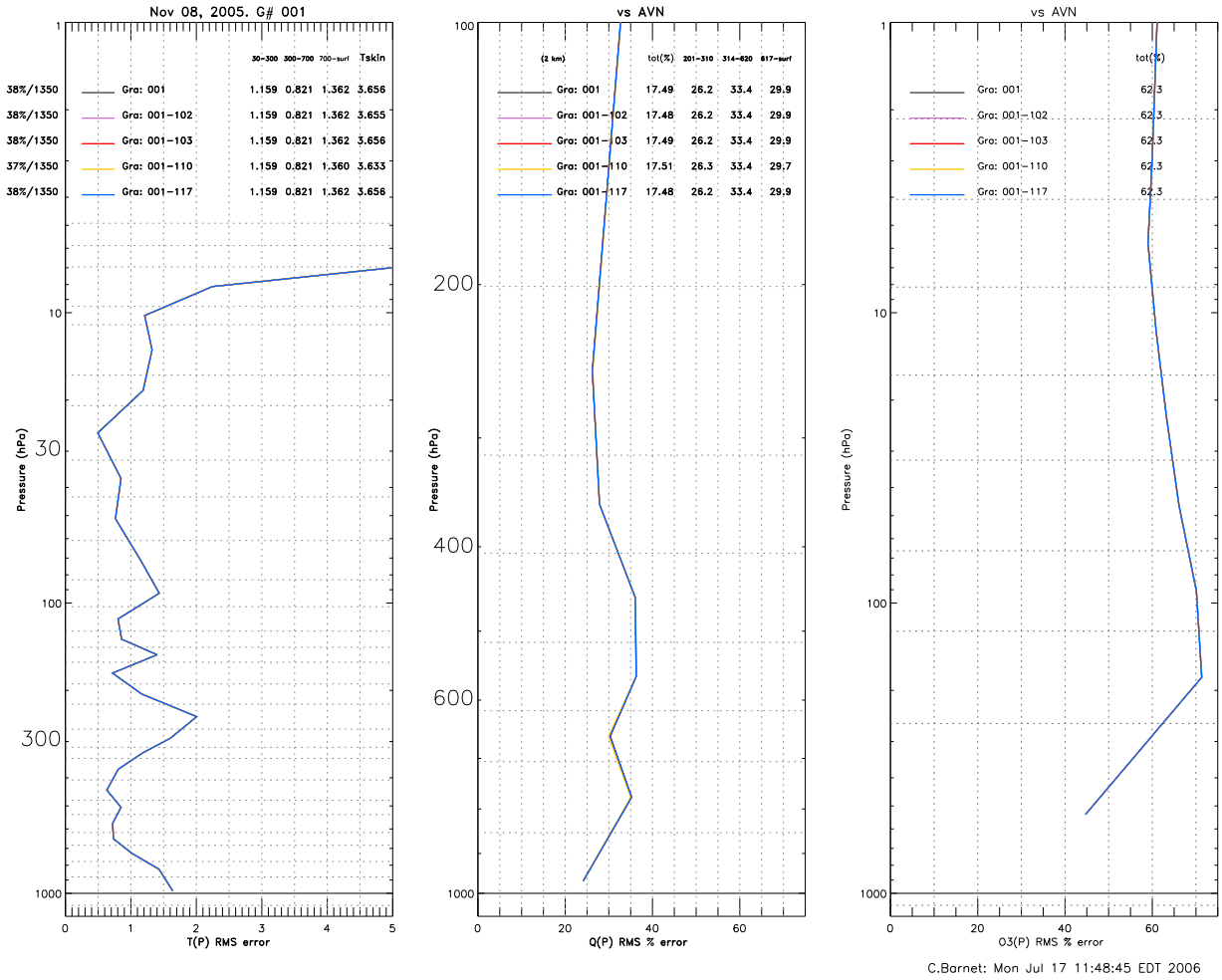


**Figure 11.** Profiles of the retrievals: left panel represents temperature, middle panel represents moisture, and right panel represents ozone. The black line is from the original, and the colored lines are from decompressed files that were corrupted during the transmission.

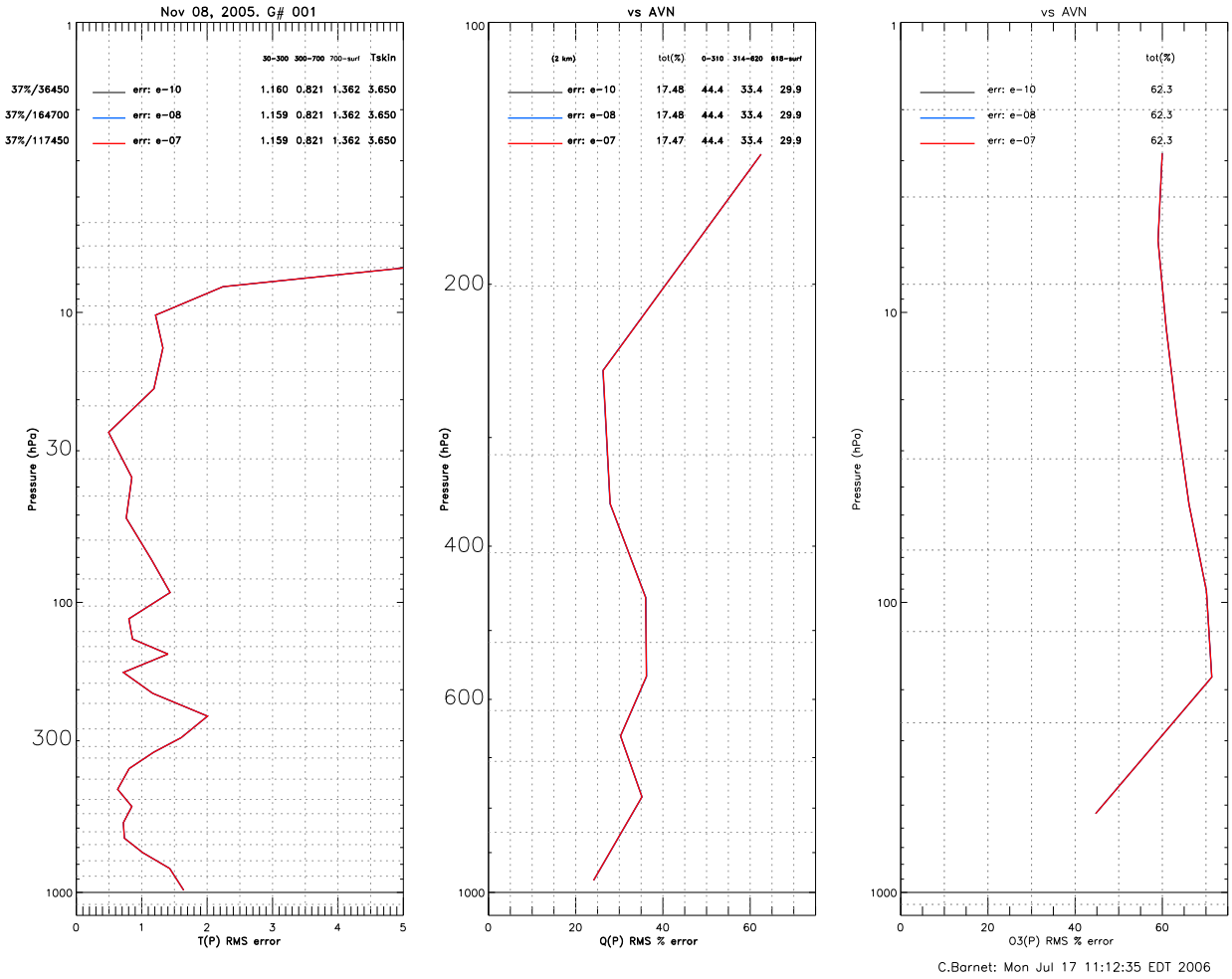


**Figure 12.** Profiles of RMS errors of the retrievals: left panel represents temperature, middle panel represents moisture, and right panel represents ozone. The black line is from the original, and the colored lines are from different error-correction cases.

these evaluations, we have implemented an end-to-end system that subjects AIRS digital counts to lossless compression, simulated transmission through a noisy channel by means of satellite forward error correcting coding, decompression, and processing though the NOAA/NESDIS/STAR near-real-time AIRS processing system, which outputs temperature, water vapor, ozone, and trace gas atmospheric retrievals. The retrievals were evaluated by NOAA scientists.

To summarize, we have developed a lossless compression technique that exploits the structure of the remote-sensing datasets and which in addition is characterized by meticulous attention to the effects of transmission noise, and performed evaluations though the NOAA/NESDIS/STAR near-real-time AIRS processing system. From our testing results, the impact of this approach on the retrieval products remains minimal across all cases, regardless of the different BER levels. More experiments and case studies are under way and further improvements of this approach are expected.

## 10. ACKNOWLEDGMENTS

This work is being sponsored by NOAA/NESDIS and has been prepared in support of the NOAA/NESDIS 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. H. L. Huang and P. Antonelli, "Application of principal component analysis to high-resolution infrared measurement compression and retrieval," *J. Appl. Meteorol.* **40**(3), pp. 365–388, 2001.
2. B. Huang, A. Ahuja, H.-L. Huang, T. Schmit, and R. Heymann, "Current status of ultraspectral sounder data compression," in *Proc. of SPIE, Satellite Data Compression, Communications, and Archiving*, **5889**, 2005.
3. B. Huang, A. Ahuja, and H.-L. Huang, "Fast precomputed VQ with optimal bit allocation for lossless compression of ultraspectral sounder data," in *Proc. of IEEE Data Compression Conference*, pp. 408–417, 2005.
4. I. Gladkova, L. Roytman, M. Goldberg, and J. Weber, "Compression of AIRS data using empirical mode decomposition," in *Proc. of SPIE*, **5548**, pp. 88–98, 2004.
5. I. Gladkova, L. Roytman, M. Goldberg, and J. Weber, "Design of a compression algorithm for GOES data," in *Third GOES-R User Conference*, (Boulder, CO), May 2004.
6. I. Gladkova, L. Roytman, and M. Goldberg, "Adaptive clustering for hyperspectral sounder data compression," in *Proc. of SPIE, Satellite Data Compression, Communications, and Archiving*, **5889**, 2005.
7. I. Gladkova, N. R. Nalli, W. W. Wolf, L. Zhou, and M. D. Goldberg, "Efficient compression of hyperspectral satellite data by adaptive clustering," *J. Atmos. Oceanic Technol.* (submitted), 2006.
8. 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.
9. E. Grayver and P. Dafesh, "Multi-modulation programmable transceiver system with turbo coding," in *IEEE Aerospace*, (Big Sky, MT), Feb. 2005.
10. C. E. Shannon, "Communications in the presence of noise," *Proc. of IRE* **37**, pp. 10–21, 1949.
11. C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal* **27**, pp. 379–423, 623–656, 1948.
12. H. H. Aumann, D. T. Gregorich, S. L. Gaiser, D. F. Hagan, T. S. Pagano, L. Strow, and D. Ting, "AIRS level lb algorithm theoretical basis document (ATBD) part 1 (IR)," (Technical Report), 10 November 2000.
13. A. Natu and D. Taubman, "Unequal protection of JPEG2000 code-streams in wireless channels," in *Proceedings of GLOBECOM'02*, 2002.
14. B. A. Banister, B. Belzer, and T. R. Fischer, "Robust Image Transmission Using JPEG2000 and Turbo-Codes," *IEEE Signal Processing Letters* **9**, April 2002.
15. 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.