IPN Progress Report 42-163 November 15, 2005

Low-Complexity Lossless Compression of Hyperspectral Imagery via Adaptive Filtering

M. Klimesh[1]

A low-complexity, adaptive predictive technique for lossless compression of hyperspectral data is presented. The technique relies on the sign algorithm from the repertoire of adaptive filtering. The compression effectiveness obtained with the technique is competitive with that of the best of previously described techniques with similar complexity.

I. Introduction

Onboard compression of hyperspectral imagery is important for reducing the burden on downlink resources. Here we describe a novel adaptive predictive technique for lossless compression of hyperspectral data. This technique uses an adaptive filtering method and achieves a combination of low complexity and compression effectiveness that is competitive with the best results from the literature. Although we are primarily interested in application to hyperspectral imagery, the technique is also generally applicable to any sort of multispectral imagery.

The algorithm described in this article seems to represent a particularly effective way of using adaptive filtering for predictive compression of hyperspectral images. However, although much analysis, experimentation, and refinement was needed to reach this point, the current state merely represents a convenient point at which to stop and document the results. There are many potential avenues for further development and improvement, some of which are mentioned in Section IV.

Estimation of sample values by linear prediction is a natural strategy for lossless compression of hyperspectral images. The differences between the estimates and the actual sample values are encoded into the compressed bitstream. This is a form of predictive compression, or, more specifically, a form of differential pulse code modulation (DPCM). Only previously encoded samples are used to predict a given sample in order that the prediction operation can be duplicated by the decoder.

We would like to have a predictor that produces estimates that are as accurate as possible. Developing a technique to do this is a central task in the overall compressor development. The primary contribution of this article is a novel prediction method that relies on a low-complexity adaptive filtering technique. Specifically, a key feature of our compressor is that it uses the sign algorithm.

---

[1] Communications Architectures and Research Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

The sign algorithm [6] is a relative of the least mean square (LMS) algorithm [21,22], a well-known low-complexity adaptive filtering algorithm. The sign algorithm is also known as the sign-error algorithm and the binary reinforcement algorithm. There are many other relatives of the LMS algorithm, and some of these may be useful for our application as well. The LMS algorithm and its relatives have found extensive application in audio compression.

There are a few reports in the literature concerning the application of the LMS algorithm to images for various filtering operations such as denoising. A straightforward extension of the LMS algorithm to two-dimensional (2-D) images is described in [8]; the article observes that it may be useful for image compression. Later the same authors analyze the effect of a nonzero mean in images and show that it has a detrimental effect on the LMS algorithm [9]. They propose normalization of filter weights to unity to alleviate the problem. Lin et al. [13] describe a method of local mean estimation and subtraction prior to use of the 2-D LMS algorithm, with application to image processing. Several variations of this strategy are compared in [25], with application to filtering magnetic resonance imaging (MRI) data.

In a few cases, researchers have been directly interested in applying the LMS algorithm to image compression. An early example occurs in [2], where the application is fixed-rate, lossy, predictive compression of 2-D images. Reference [4] contains an example of the application of the LMS algorithm to lossless predictive image compression. Reference [12] describes the use of a three-dimensional (3-D) LMS algorithm for restoration of and prediction in hyperspectral images (but provides few details).

There has been a fair amount of work on lossless predictive compression of hyperspectral images that does not involve the LMS algorithm or its relatives. The most recent and relevant of these include [1,14,16,18]. In particular, the methods used by Rizzo et al. [18] have low complexity and yield a compression effectiveness similar to that of our methods. The best compression-effectiveness results reported in the literature may be from [1], but those results are obtained with methods of moderately high complexity.

II. Algorithm Description

The essence of our hyperspectral compression algorithm is adaptive linear predictive compression using the sign algorithm for filter adaptation, with local mean estimation and subtraction.

We start with a brief description of the LMS algorithm and the sign algorithm. For both of these algorithms, a desired signal d_k is to be estimated from an input (column) vector $\mathbf{u}_k$. Here k is an index that increases sequentially. The estimate $\hat{d}_k$ is a linear function of $\mathbf{u}_k$; specifically, $\hat{d}_k = \mathbf{w}_k^T \mathbf{u}_k$, where $\mathbf{w}_k$ is the filter weight vector at index k.

After an estimate $\hat{d}_k$ is made, the error between the estimate and the desired signal is computed. Specifically, $e_k = \hat{d}_k - d_k$. This error value is used to update the filter weights. For the LMS algorithm,

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \mathbf{u}_k e_k$$

For the sign algorithm,

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \mathbf{u}_k \operatorname{sgn}(e_k)$$

In each case, μ is a positive, scalar parameter (the step size parameter) that controls the trade-off between convergence speed and average steady-state error. A small μ results in better steady-state performance but slower convergence. In some variants of these algorithms, the value of μ changes over time.

The sign algorithm has the property that under certain general assumptions the weight vectors it produces become clustered around the optimum weight vector in terms of minimizing the mean absolute estimation error. For a sufficiently small adaptation step size parameter, the asymptotic mean absolute estimation error can be made to be as close as desired to the minimum possible [6].

A straightforward method of applying the sign algorithm or the LMS algorithm to the prediction step in image compression is to identify d_k with an image sample to be estimated, and $\mathbf{u}_k$ with a causal neighborhood of the image sample. For example, in a hyperspectral image, let $s(x, y, z)$ be the sample value at spatial location (x, y) in spectral band z. To estimate a sample from the three previously encoded samples that are adjacent in the three dimensions, we could apply the LMS or sign algorithm in such a way that $d_k = s(x, y, z)$ corresponds to $\mathbf{u}_k = [s(x-1, y, z), s(x, y-1, z), s(x, y, z-1)]^T$. Unfortunately, this method does not work well, yielding poor combinations of convergence speed and steady-state performance. We had some success combating this problem by normalizing filter weights to sum to unity (after scaling by spectral band signal levels), a technique that is closely related to a technique suggested in [9]. However, we eventually settled on a local mean subtraction method motivated by [13].

In our local mean subtraction method, for each sample we compute a preliminary estimate using a fixed, causal, linear predictor involving only samples from the same band. Denote the preliminary estimate of sample $s(x, y, z)$ by $\tilde{s}(x, y, z)$. The desired signal in the LMS or sign algorithm is now defined as $d_k = s(x, y, z) - \tilde{s}(x, y, z)$. For our example of an estimate from the three adjacent samples, we use

$$\mathbf{u}_k = \begin{bmatrix} s(x-1, y, z) - \tilde{s}(x, y, z) \\ s(x, y-1, z) - \tilde{s}(x, y, z) \\ s(x, y, z-1) - \tilde{s}(x, y, z-1) \end{bmatrix}$$

as the corresponding input vector. The general rule is to adjust each sample in the prediction neighborhood by the preliminary estimate in the same band as the sample but at the spatial location of the sample being predicted. Since this is done as part of a predictive compression algorithm, the difference $\hat{d}_k - d_k$ is encoded in the compressed bitstream. The decompressor decodes this difference from the bitstream and can compute $\hat{d}_k$ and $\tilde{s}(x, y, z)$ from previously decoded samples, and therefore can reconstruct the value $s(x, y, z)$.

We note that our local mean subtraction step is reminiscent of the transform step in the transform domain LMS algorithm [3,17]. This connection may warrant further exploration.

A. Algorithm Specifics

Implementation of the above predictive compression framework involves many choices. Here we describe the specifics of the algorithm that we used to generate our test results. Many other combinations of choices are possible.

Conceptually, an image is partitioned spatially into conveniently sized fixed regions, and within each region the spectral bands are compressed sequentially, with each spectral band compressed in its entirety before moving on to the next band. The predictor statistics are reset with each new band. In practice, the data can be compressed in the order they are acquired, maintaining separate statistics for each band and switching among them as necessary. In either case, within a band, samples are processed in raster scan order. In our tests, the regions are slices of a fixed height, namely 32, and each region is compressed independently. The independent compression is done both to provide a means of limiting the effects of data loss in an onboard implementation (error containment) and as a convenience allowing the entire region to reside in memory during compression and decompression in our tests.

We use a six sample prediction neighborhood with three samples from the same band as the sample to be predicted, and one sample each from the three preceding bands. Specifically, the prediction neighborhood consists of the samples at coordinates $(-1, 0, 0)$, $(-1, -1, 0)$, $(0, -1, 0)$, $(0, 0, -1)$, $(0, 0, -2)$, and $(0, 0, -3)$ relative to the sample to be predicted, so that

$$\mathbf{u}_k = \begin{bmatrix} s(x-1, y, z) - \tilde{s}(x, y, z) \\ s(x-1, y-1, z) - \tilde{s}(x, y, z) \\ s(x, y-1, z) - \tilde{s}(x, y, z) \\ s(x, y, z-1) - \tilde{s}(x, y, z-1) \\ s(x, y, z-2) - \tilde{s}(x, y, z-2) \\ s(x, y, z-3) - \tilde{s}(x, y, z-3) \end{bmatrix}$$

We have not yet carried out any significant amount of experimentation with alternate prediction neighborhoods. For the first spectral band, the last three elements of the neighborhood are dropped so that offsets do not refer to negative band indices. Similarly, the prediction neighborhood is appropriately reduced for the second and third spectral bands. Within a band, for prediction neighborhood offsets that are outside the image bounds, the nearest valid causal sample is used. The first sample of each band of each region is simply included directly in the compressed bitstream. Within each band, compression proceeds in raster scan order.

The prediction weights are initialized to be uniform among the neighborhood, summing to 1. For the first line, μ is set to 0.00008. After each of the first 10 lines, μ is multiplied by 0.75. We chose this sequence of μ values because it seemed to produce good results; however, there is some robustness in that moderate variations to this schedule still produce good results.

Our preliminary sample estimates are produced by averaging the four nearest causal samples from the same band, namely, those at offsets $(-1, 0)$, $(-1, -1)$, $(0, -1)$, and $(1, -1)$.

The difference $\hat{d}_k - d_k$ is encoded by applying a mapping that produces a nonnegative integer and encoding this integer using Golomb codes [5,7] with parameters that are powers of two (also known as Golomb-Rice codes). This overall difference encoding procedure is very similar to that used by LOCO-I/JPEG-LS, described in [20].

In more detail, the encoding is as follows. In general, $\hat{d}_k$ is not an integer. The possible values of d_k are labeled with nonnegative integers based on how close they are to $\hat{d}_k$: the nearest is labeled 0, the next nearest is labeled 1, and so on. The label corresponding to the actual value of d_k is encoded using a Golomb code. Equivalently, let $\text{round}(\hat{d}_k)$ be the nearest integer to $\hat{d}_k$; let $\Delta_k = d_k - \text{round}(\hat{d}_k)$; and define a function f from the integers to the nonnegative integers by

$$f(n) = \begin{cases} 2n & \text{if } n \geq 0 \\ -2n - 1 & \text{if } n < 0 \end{cases}$$

Then the value to be encoded using a Golomb code is $f(\Delta_k)$ or $f(-\Delta_k)$, depending on whether $\hat{d}_k$ is less than or greater than $\text{round}(\hat{d}_k)$.

The Golomb code parameter is determined by a running estimate of the average magnitude of the Δ_k. Specifically, if a running tally includes n samples with a total Δ_k magnitude sum of s, then the Golomb code parameter is chosen to be 2^m, where m is the smallest nonnegative integer for which $n \cdot 2^m > s$. This is essentially the same as the Golomb code parameter selection mechanism of LOCO-I/JPEG-LS, as described in [20].

III. Results

We have tested our compressor on several datasets from the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). These datasets include five 1997 calibrated radiance datasets available from the AVIRIS Web site [10]; a 2001 calibrated radiance dataset with imagery from Arizaro, Argentina (flight f010207t01, run p02r06, 11 scenes); a 2001 uncalibrated (raw) dataset with imagery from the Island of Hawaii, Hawaii (flight f011020t01, run p03r05); and a 2003 uncalibrated dataset with imagery from Maine (flight f030828t01, run p00r05).

The uncalibrated datasets each contain many scan lines at the beginning and end of the run that do not seem to contain meaningful image data. Our compression tests use consecutive 512-line scenes from the middle of the datasets; these were chosen to include the good data but are otherwise somewhat arbitrary. For the 2001 Hawaii dataset, we discard 436 lines from the top and 415 lines from the bottom, leaving six 512-line scenes. For the 2003 Maine dataset, we discard 362 lines from the top and 1944 lines from the bottom, leaving thirteen 512-line scenes. Note that we have not obtained the corresponding calibrated datasets, so the scenes as we have defined them do not necessarily match up with the scenes in the corresponding calibrated datasets. There may be some value to a comparison of results in uncalibrated and corresponding calibrated scenes.

All scenes from all datasets contain 512 lines and 224 bands. The 2003 Maine dataset scenes contain 680 samples/line, and all others contain 614 samples/line.

Table 1 contains results for the 1997 datasets. Our algorithm is labeled "fast lossless." The "ICER-3D" column contains lossless compression results for ICER-3D, a 3-D-wavelet-based compressor described in [11] and in a paper in preparation;[2] ICER-3D can be used for lossy or lossless compression. The other results for comparison are JPEG-LS [20] applied to the spectral bands independently;[3] the Rice compressor used in the Universal Source Encoder for Space (USES) chip[4] using the multispectral predictor option[5] mentioned in [19]; JPEG-LS applied to the differences between the successive spectral bands; and two versions of spectral-oriented least squares (SLSQ) [18]. SLSQ and SLSQ-OPT are relatively simple predictive compression algorithms that are based on different principles than our compressor and that use arithmetic coding; their complexity is roughly similar to that of our fast lossless compressor. The differential JPEG-LS, SLSQ, and SLSQ-OPT results were obtained from the authors of [18] and correspond to aggregate results presented in that publication.

In Table 2, we compare our fast lossless algorithm to three different compressors of moderate complexity. The 3-D CALIC compressor [24] is a nontrivial extension of the basic (2-D) context-based, adaptive, lossless image codec (CALIC) [23] algorithm to multispectral imagery; in these results, in the compression of a given band, the preceding band is used as the reference band. The M-CALIC (multiband CALIC) compressor [14] is another extension of CALIC to multispectral imagery, tailored toward exploiting the high interband correlations of hyperspectral datasets. The last column contains results for a compressor described in [1]; it is called adaptive selection of adaptive predictors (ASAP) in [14] and is more computationally intensive than any of the other compressors mentioned in this article. Table 2 contains results for four of the 1997 datasets, in each case for only the first 256 lines of the first scene because that portion of the data is used in the results given in [14].

---

[2] A. Kiely, H. Xie, M. Klimesh, and N. Aranki, "ICER-3D: A Progressive Wavelet-Based Compressor for Hyperspectral Images," in preparation for submittal to *The Interplanetary Network Progress Report*.

[3] In particular, we used version 1.1 of the JPEG-LS implementation produced by Ismail R. Ismail and Faouzi Kossentini of the Department of Electrical and Computer Engineering, University of British Columbia. Prior to compression, bands containing negative samples had their values translated by a constant sufficient to make all the values positive.

[4] The Rice/USES results were obtained with block length $J = 16$; this choice gives the best results, but the fact that the scene widths are not a multiple of 16 seems to cost about 0.05 bits/sample for the datasets of our tests.

[5] The multispectral option uses a fixed, internally computed, 2-D predictor using the spectral dimension and one spatial dimension. However, we note that the USES chip allows arbitrary predictors provided they are computed externally.

Table 1. Bit rates achieved for compression of scenes from the calibrated 1997 AVIRIS datasets. Results are given in bits/sample.

Scene	Fast lossless	ICER-3D	JPEG-LS (2-D)	Rice/USES multispectral	Differential JPEG-LS	SLSQ	SLSQ-OPT
Cuprite 1	4.89	5.14	7.13	6.00	5.44	5.03	4.90
Cuprite 2	5.02	5.34	7.50	6.13	5.58	5.09	4.97
Cuprite 3	4.92	5.16	7.16	6.00	5.45	5.06	4.92
Cuprite 4	4.98	5.21	7.16	6.05	5.51	5.10	4.96
Jasper Ridge 1	5.04	5.41	7.72	6.17	5.62	5.06	4.95
Jasper Ridge 2	5.02	5.37	7.67	6.12	5.59	5.05	4.94
Jasper Ridge 3	5.07	5.47	7.90	6.19	5.67	5.10	4.99
Jasper Ridge 4	5.07	5.47	7.87	6.22	5.67	5.11	5.00
Jasper Ridge 5	5.02	5.39	7.75	6.14	5.60	5.06	4.94
Low altitude 1	5.37	5.70	7.81	6.53	5.97	5.38	5.30
Low altitude 2	5.42	5.76	7.95	6.58	6.02	5.40	5.33
Low altitude 3	5.30	5.58	7.57	6.42	5.88	5.33	5.23
Low altitude 4	5.32	5.58	7.53	6.42	5.89	5.37	5.26
Low altitude 5	5.37	5.63	7.60	6.47	5.91	5.40	5.30
Low altitude 6	5.29	5.56	7.52	6.42	5.85	5.34	5.24
Low altitude 7	5.29	5.60	7.64	6.43	5.88	5.34	5.24
Lunar Lake 1	4.99	5.19	6.98	6.02	5.49	5.12	4.99
Lunar Lake 2	4.94	5.14	6.96	5.97	5.44	5.07	4.93
Moffett Field 1	5.12	5.48	7.78	6.24	5.70	5.15	5.03
Moffett Field 2	5.11	5.40	7.57	6.20	5.60	5.08	4.98
Moffett Field 3	4.98	5.12	7.03	5.96	5.41	4.96	4.86
Average	5.12	5.41	7.51	6.22	5.68	5.17	5.06

Table 2. Bit rates achieved for compression of the first half-scenes (256 lines) from four of the calibrated 1997 AVIRIS datasets. Results are given in bits/sample.

Dataset	Fast lossless	3D-CALIC	M-CALIC	ASAP
Cuprite	4.86	5.23	4.97	4.87
Jasper Ridge	5.02	5.20	5.05	4.83
Lunar Lake	5.02	5.17	4.88	4.76
Moffett Field	5.06	4.92	4.73	4.60
Average	4.99	5.13	4.91	4.76

Table 3 contains results for the Arizaro dataset, and Table 4 contains results for the two uncalibrated datasets.

Although we do not have any direct comparisons, it appears from Tables 1 through 4 that the uncalibrated datasets compress much better than the calibrated datasets. This may seem surprising at first, since the inherent information content of the calibrated datasets should not be appreciably higher (if at all higher) than that of the uncalibrated datasets because the calibrated datasets are derived from the uncalibrated datasets. However, calibration introduces redundancy of a type that does not occur in natural images and would require specialized techniques to exploit. In particular, when calibration increases the dynamic range of a spectral band, the least significant bits of the samples typically contain redundancy that is not exploited. Since our compressor is intended for eventual use on uncalibrated data, we have not attempted to exploit the redundancy introduced by calibration (and we believe the same applies to the other compressors in the tables).

IV. Potential Improvements

As we mentioned earlier, the algorithm described in this article seems to represent a particularly effective way of using adaptive filtering for predictive compression of hyperspectral images, but the current state merely represents a convenient point at which to stop and document the results. We now mention a few potential avenues for further development and improvement.

A fairly obvious step would be to investigate the effect of modifying the prediction neighborhood. For example, a larger neighborhood might give more accurate estimates, but would increase complexity and could reduce adaptation speed. Another direction to pursue would be to incorporate some form of context modeling: different sets of statistics could be used depending on the behavior of samples in a local causal neighborhood. These statistics could include prediction weight vectors and/or the mean absolute error values used for choosing the Golomb code parameters.

The prediction accuracy might be improved by tweaking the sign algorithm parameters or by changing the adaptive filtering technique more extensively. Changing the adaptation step size parameter schedule is a possible minor parameter change. Examples of larger changes include using either another variation

Table 3. Bit rates achieved for compression of scenes from the calibrated 2001 Arizaro dataset. Results are given in bits/sample.

Scene	Fast lossless	ICER-3D	JPEG-LS (2-D)	Rice/USES multispectral
2001 Arizaro 1	4.54	4.54	5.76	5.55
2001 Arizaro 2	4.51	4.49	5.71	5.51
2001 Arizaro 3	4.49	4.46	5.65	5.48
2001 Arizaro 4	4.50	4.49	5.71	5.52
2001 Arizaro 5	4.52	4.57	5.88	5.51
2001 Arizaro 6	4.54	4.64	6.12	5.52
2001 Arizaro 7	4.61	4.62	5.91	5.60
2001 Arizaro 8	4.67	4.68	6.01	5.65
2001 Arizaro 9	4.82	4.97	6.67	5.78
2001 Arizaro 10	4.61	4.70	6.11	5.59
2001 Arizaro 11	4.56	4.60	5.98	5.55
Average	4.58	4.62	5.95	5.57

Table 4. Bit rates achieved for compression of scenes from the uncalibrated 2001 Hawaii and 2003 Maine AVIRIS datasets. Results are given in bits/sample.

Scene	Fast lossless	ICER-3D	JPEG-LS (2-D)	Rice/USES multispectral
2003 Maine 1	2.92	3.38	5.00	4.02
2003 Maine 2	2.89	3.33	4.88	3.98
2003 Maine 3	2.98	3.49	5.21	4.12
2003 Maine 4	2.93	3.41	5.01	4.07
2003 Maine 5	2.86	3.27	4.70	3.93
2003 Maine 6	2.81	3.21	4.59	3.90
2003 Maine 7	2.79	3.18	4.54	3.87
2003 Maine 8	2.77	3.19	4.60	3.87
2003 Maine 9	2.84	3.28	4.75	3.95
2003 Maine 10	2.82	3.23	4.66	3.88
2003 Maine 11	2.77	3.19	4.56	3.85
2003 Maine 12	2.73	3.15	4.49	3.82
2003 Maine 13	2.80	3.24	4.68	3.89
2001 Hawaii 1	2.75	3.12	4.93	3.77
2001 Hawaii 2	2.85	3.32	5.19	4.00
2001 Hawaii 3	2.86	3.34	5.11	4.03
2001 Hawaii 4	2.79	3.17	4.70	3.89
2001 Hawaii 5	2.71	3.06	4.44	3.79
2001 Hawaii 6	2.46	2.72	3.79	3.39
Average	2.81	3.23	4.73	3.89

of the sign algorithm or an altogether different algorithm from the LMS family. In addition, there may be better mappings from the hyperspectral image data to the input of the sign algorithm as compared with our method, which uses preliminary estimates. Changing the way prediction weights are initialized could easily result in worthwhile improvements in compression effectiveness. A simple initialization that is different from what is currently used might accomplish this. In some scenarios, it may be reasonable to initialize the prediction weights to carefully chosen values that depend on the spectral band. A somewhat related point is that changing the amount of data per independently coded region also has an effect on compression effectiveness.

Finally, the efficiency of the entropy coding of the prediction errors could be improved. Judging from redundancy plots presented in [15], we estimate our results would improve by roughly 0.05 bits/sample if we were to use arithmetic coding; however, the cost of this would be some increase in complexity.

Acknowledgments

We thank Robert Green and others from the AVIRIS project for providing us with the 2001 Arizaro dataset and the uncalibrated datasets for our compression tests.

We thank the authors of [18], especially Francesco Rizzo and Giovanni Motta, for providing us with the individual scene data corresponding to the aggregate results reported in [18].

We thank Pen-Shu Yeh for providing us with USES emulation software and information on using its multispectral compression option. Mark Wilde generated the Rice/USES results using the software. Mark Wilde also generated the JPEG-LS results. The ICER-3D results were generated by Hua Xie. Frank Zee assisted with some of the compression tests that preceded our current results.

References

[1] B. Aiazzi, L. Alparone, and S. Baronti, "Near-Lossless Compression of 3-D Optical Data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 11, pp. 2547–2557, November 2001.

[2] S. Alexander and S. Rajala, "Image Compression Results Using the LMS Adaptive Algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, no. 3, pp. 712–714, June 1985.

[3] F. Beaufays, "Transform-Domain Adaptive Filters: An Analytical Approach," *IEEE Transactions on Signal Processing*, vol. 43, no. 2, pp. 422–431, February 1995.

[4] R. Ding and L. Wu, "Enhanced DPCM Using LMS Predictor and Composite Source Model," *Electronics Letters*, vol. 33, no. 9, pp. 740–741, April 24, 1997.

[5] R. G. Gallager and D. C. Van Voorhis, "Optimal Source Codes for Geometrically Distributed Integer Alphabets," *IEEE Transactions on Information Theory*, vol. IT-21, no. 2, pp. 228–230, March 1975.

[6] A. Gersho, "Adaptive Filtering with Binary Reinforcement," *IEEE Transactions on Information Theory*, vol. IT-30, no. 2, pp. 191–199, March 1984.

[7] S. W. Golomb, "Run-Length Encodings," *IEEE Transactions on Information Theory*, vol. IT-12, no. 3, pp. 399–401, July 1966.

[8] M. M. Hadhoud and D. W. Thomas, "The Two-Dimensional Adaptive LMS (TDLMS) Algorithm," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 5, pp. 485–494, May 1988.

[9] M. M. Hadhoud and D. W. Thomas, "The Effect of the Image Local Mean on the Two-Dimensional Least Mean Square Algorithm Weight Convergence," *Journal of Modern Optics*, vol. 36, no. 4, pp. 545–549, April 1989.

[10] AVIRIS Web site, http://aviris.jpl.nasa.gov/

[11] M. Klimesh, A. Kiely, H. Xie, and N. Aranki, "Spectral Ringing Artifacts in Hyperspectral Image Data Compression," *The Interplanetary Network Progress Report*, vol. 42-160, Jet Propulsion Laboratory, Pasadena, California, pp. 1–17, February 15, 2005. http://ipnpr/progress_report/42-160/160C.pdf

[12] M. Larsen, T. Bose, and A. Venkatachalam, "Hyperspectral Image Restoration and Coding," *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, vol. 2, Pacific Grove, California, pp. 1740–1744, November 3–6, 2002.

[13] J. N. Lin, X. Nie, and R. Unbehauen, "Two-Dimensional LMS Adaptive Filter Incorporating a Local-Mean Estimator for Image Processing," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. 40, no. 7, pp. 417–428, July 1993.

[14] E. Magli, G. Olmo, and E. Quacchio, "Optimized Onboard Lossless and Near-Lossless Compression of Hyperspectral Data Using CALIC," *IEEE Geoscience and Remote Sensing Letters*, vol. 1, no. 1, pp. 21–25, January 2004.

[15] N. Merhav, G. Seroussi, and M. J. Weinberger, "Optimal Prefix Codes for Sources with Two-Sided Geometric Distributions," *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 121–135, January 2000.

[16] J. Mielikainen, A. Kaarna, and P. Toivanen, "Lossless Hyperspectral Image Compression via Linear Prediction," *Proceedings of Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery VIII, SPIE 4725*, S. S. Shen and P. E. Lewis, eds., Orlando, Florida, pp. 600–608, April 1–5, 2002.

[17] S. Narayan, A. Peterson, and M. Narasimha, "Transform Domain LMS Algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, no. 3, pp. 609–615, June 1983.

[18] F. Rizzo, B. Carpentieri, G. Motta, and J. A. Storer, "Low-Complexity Lossless Compression of Hyperspectral Imagery via Linear Prediction," *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 138–141, February 2005.

[19] J. Venbrux, J. Gambles, D. Wiseman, G. Zweigle, W. H. Miller, and P.-S. Yeh, "A VLSI Chip Set Development for Lossless Data Compression," *Ninth AIAA Computing in Aerospace Conference*, San Diego, California, October 19–21, 1993.

[20] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, August 2000.

[21] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. C. Goodlin, J. R. Zeidler, R. H. Hearn, and E. Dong, "Adaptive Noise Cancelling: Principles and Applications," *The Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, December 1975.

[22] B. Widrow and M. E. Hoff, Jr., "Adaptive Switching Circuits," *IRE Western Electric Show and Convention Record*, Pt. 4, Los Angeles, California, pp. 96–104, August 23–26, 1960.

[23] X. Wu and N. Memon, "Context-Based, Adaptive, Lossless Image Coding," *IEEE Transactions on Communications*, vol. 45, no. 4, pp. 437–444, April 1997.

[24] X. Wu and N. Memon, "Context-Based Lossless Interband Compression—Extending CALIC," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 994–1001, June 2000.

[25] K. Ying, B. D. Clymer, and P. Schmalbrock, "Adaptive Filtering for High Resolution Magnetic Resonance Images," *Journal of Magnetic Resonance Imaging*, vol. 6, no. 2, pp. 367–377, March/April 1996.