Lossless Hyperspectral-Image Compression Using Context-Based Conditional Average

Hongqiang Wang, S. Derin Babacan, and Khalid Sayood, *Senior Member, IEEE*

***Abstract*—In this paper, a new algorithm for lossless compression of hyperspectral images is proposed. The spectral redundancy in hyperspectral images is exploited using a context-match method driven by the correlation between adjacent bands. This method is suitable for hyperspectral images in the band-sequential format. Moreover, this method compares favorably with the recent proposed lossless compression algorithms in terms of compression, with a lower complexity.**

***Index Terms*—Conditional average, context coding, correlation, entropy code, Golomb–Rice code, hyperspectral image, image coding.**

I. Introduction

CONCERNED with the effects of global climate change, the National Aeronautics and Space Administration (NASA) initiated the Mission to Planet Earth Enterprise in 1991. Many efforts of NASA and other national space agencies are coming to fruition. Among these efforts has been the development of remote-sensing instruments with high levels of spatial and spectral resolutions. The products generated by these sensors promise to revolutionize our understanding of climatology, meteorology, and land management. As the amount of data generated by these sensors is enormous and the number of sensors continues to grow, it is clear that the role of data compression will be crucial in this development. At different points in the path from the sensor to the end-user, the compression needs are different, and both lossy and lossless compression approaches are needed.

Lossy compression has been an enabling technology for the multimedia revolution of the past decades, in part because the human visual and hearing system are insensitive to certain kinds and levels of distortion. Thus, selective loss of information can be used to significantly enhance the compression performance. However, hyperspectral images are not solely intended to be viewed by human beings. They are usually processed for various applications such as automatic feature extraction, classification, target detection, and object identification. Distortion from lossy compression might be amplified during the processing and, therefore, have a significant deleterious effect. As such, lossy compression is not always desirable. At a minimum, lossless compression is needed if the data are to be stored for some period of time on the acquisition platform and for transmission from the acquisition platform to the ground station. Lossless compression is also needed for data archiving. Our focus in this paper is on lossless compression.

Compression relies on statistical structure in the data. In hyperspectral images, there are basically two types of correlation available. One is spatial correlation in adjacent pixels in the same band, and the other is spectral correlation in pixels between adjacent bands. The spatial correlation can be easily exploited using general compression techniques developed for gray-scale images. However, the solution of how to efficiently explore the redundancy between adjacent bands for high-resolution hyperspectral images is not yet well established.

Most lossless hyperspectral-image compression techniques can be classified as vector quantization (VQ) techniques, predictive coding techniques, or transform coding techniques. Ryan and Arnold [1] discuss various VQ techniques and propose a mean-normalized VQ (M-NVQ). In [2], Pickering and Ryan use a discrete cosine transform in both the spatial and spectral domains to exploit the redundancy in the M-NVQ output. In [3], Motta *et al.* partition the input vectors into a number of consecutive subsegments and use a variation of the generalized Lloyd algorithm to train vector quantizers for these subsegments. The subsegments are then quantized independently, and the quantization residual is entropy coded to achieve lossless compression. This method is further optimized in [4] by minimizing the distortion caused by local partition boundaries. The optimized method is called LPVQ. VQ-based approaches usually require offline codebook training and online quantization-index searching. This makes them computationally expensive and not always well suited for real-time applications.

Predictive coding approaches usually employ spatial, spectral, and hybrid predictors to decorrelate the image, followed by entropy coders [5]–[13]. In [7], Roger and Cavenor use an optimum linear predictor [in the minimum mean-square error (mmse) sense] and entropy code the residual. In [8], Aiazzi *et al.* use fuzzy clustering and fuzzy prediction to select a number of predictors for data decorrelation in both the spatial and spectral domains, followed by a context-based arithmetic coder. In [9], Wu and Memon extend the CALIC algorithm [19] from 2-D to 3-D. The algorithm switches between two modes, intraband and interband prediction, based on the correlation coefficient. The resulting residual is then coded using context-based arithmetic codes. In [10], Magli *et al.* propose a modified 3-D CALIC (M-CALIC) for lossless and near-lossless compression. Rather than switching between

Manuscript received October 25, 2006; revised May 15, 2007. This work was supported in part by the NASA Goddard Space Flight Center under a grant.

H. Wang and K. Sayood are with the Department of Electrical Engineering, University of Nebraska–Lincoln, Lincoln, NE 68503 USA (e-mail: hqwang@eecomm.unl.edu; ksayood@eecomm.unl.edu).

S. D. Babacan is with the Department of Electrical Engineering, Northwestern University, Evanston, IL 60208-3118 USA (e-mail: sdb@northwestern.edu).

Digital Object Identifier 10.1109/TGRS.2007.906085

interband and intraband modes, as in 3-D CALIC, M-CALIC works with a full interband and a universal spectral predictor. It refines the prediction models and optimizes parameters of context coding to achieve a good coding performance. In [11], Rizzo *et al.* propose Spectral-oriented Least SQuares (SLSQ), in which spectral correlation is exploited using linear prediction, and the prediction error is then entropy coded. Slyz and Zhang [12] propose a fast compression algorithm with random-access capability, in which a predesigned code based on the residual error from interband prediction is applied and predictor coefficients and errors are encoded. In [13], Rizzo *et al.* propose a low-complexity method using two predictors: one is an interband linear predictor, and the other is an interband least square predictor. The compression ratio is further optimized at a moderate cost of more memory and delay. In [14], Mielikainen and Toivanen propose a clustered differential pulse-code modulation scheme, where the spectral bands of the image are clustered then filtered using an optimum linear filter for each cluster. The output of the filters is encoded using an adaptive entropy coder. In their follow-up work in [15], which they call Look-Up Tables (LUTs), the pixel that is nearest and equal to the pixel colocated with the one to be predicted in the previous band is taken as the prediction. The sign of the residual is coded first, followed by adaptive range coding of its absolute value. Predictive coding schemes, so far, seem to have the best compression performance, with the added advantage that their computational cost and complexity are usually less than VQ-based methods and, therefore, promising for onboard application. In [16], Penna *et al.* apply a 3-D wavelet transform to decorrelate the image in the spectral and spatial domains and develop a progressive coding scheme that works from lossy to lossless coding and complies with the second part of the JPEG 2000 Standard. Transform-based schemes can yield excellent coding gain for lossy compression at low bit rates while their lossless coding performance is inferior to these specialized lossless compression schemes. In [17], Tate shows that the band order could be rearranged and optimized for better interband prediction, and therefore, better compression ratios could be achieved. Recently, Zhang and Liu [18] proposed a two-step adaptive spectral-band-reordering algorithm. First, the bands are classified into groups based on the correlation factor of adjacent bands, and then, a reordering algorithm based on the Prim algorithm is applied to each group. A prediction method called ABPCNEF is proposed to take advantage of the similarity of structure and pixel relationship between two neighboring spectral bands, and the residual is coded using adaptive arithmetic coding.

We propose a lossless compression method which consists of context-based conditional average prediction (CCAP) followed by entropy coding. Our primary goal is to develop a fast lossless compression algorithm suitable for real-time implementation. The coding algorithm involves interband and intraband prediction, context match, CCAP, and entropy coding. Simulation results show that our method achieves competitive compression ratios with low complexity and computational cost. This paper is organized as follows. The compression method is presented in Section II. The results are given in Section III, and in Section IV, we conclude this paper.

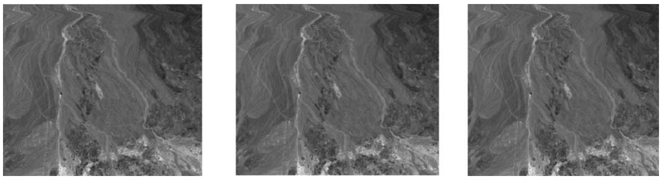

Fig. 1. Cuprite scene 1 with 101st, 102nd, and 103rd bands.

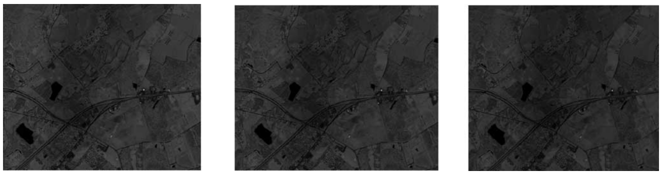

Fig. 2. Low-altitude scene 1 with 101st, 102nd, and 103rd bands.

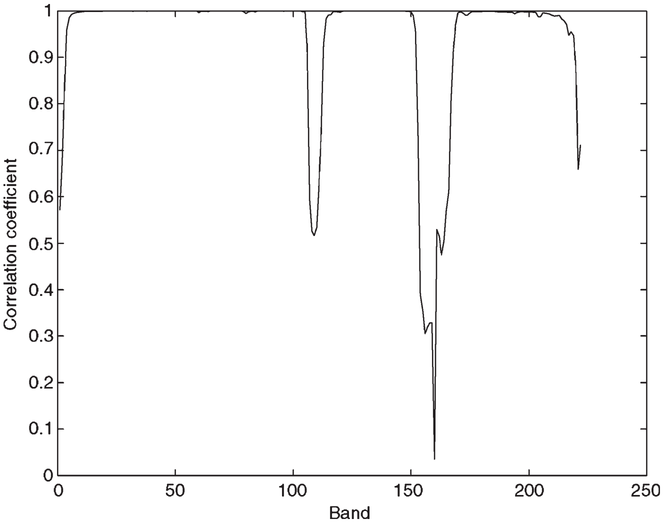

Fig. 3. Correlation coefficient between two adjacent bands of Cuprite scene 1.

II. Proposed Algorithm

A. Correlation Coefficient and Intraband Prediction

A salient property of hyperspectral images is that strong spectral correlation exists throughout almost all bands. This is shown in Figs. 1 and 2, which show the 101st, 102nd, and 103rd bands of the Cuprite scene 1 and low-altitude scene 1, respectively. Given two blocks, A and B, which are spatially located in the same place in two neighboring bands, their spectral correlation can be described by the correlation coefficient defined as follows:

$$\rho(\mathrm{A}, \mathrm{B}) = \frac{\sum_{i=1}^{m}\left[(a_i - \bar{a})(b_i - \bar{b})\right]}{\sqrt{\sum_{i=1}^{m}(a_i - \bar{a})^2 \sum_{i=1}^{m}(b_i - \bar{b})^2}} \tag{1}$$

where a_i and b_i denote the individual pixels in blocks A and B, respectively, $\bar{a}$ and $\bar{b}$ represent the mean of blocks A and B, respectively, and m is the total number of pixels in one block. Fig. 3 shows the correlation between two consecutive bands of Cuprite scene 1, from the first band to the 224th band. For most bands, the correlation value $\rho(\mathrm{A}, \mathrm{B})$ is close to one, although the dynamic range of pixels can be large and different

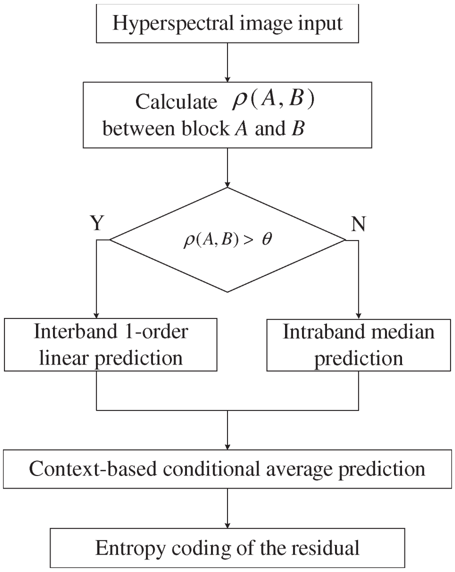
Hyperspectral image input
Calculate $\rho(A,B)$ between block A and B
Y
$\rho(A,B) > \theta$
N
Interband 1-order linear prediction
Intraband median prediction
Context-based conditional average prediction
Entropy coding of the residual

Fig. 4. Flowchart of the proposed scheme.

in neighboring bands. This implies that a first-order linear predictor using a scaling factor and an offset may work well to model the relationship between these bands. Another fact evident from Fig. 3 is that, in certain bands, the pixel intensity decreases significantly and the correlations between these bands are very weak. Examples of such bands are the bands from 104th to 110th for Cuprite image 1. Other hyperspectral images also exhibit similar behavior. This could be because, in these bands, the signal associated with these frequencies is greatly attenuated by the atmosphere or the materials being imaged. Therefore, noise dominates in these bands.

We propose a two-stage compression algorithm, as shown in Fig. 4. Note that this method is suitable for hyperspectral images in the band-sequential format. The first stage attempts to decorrelate the highly correlated bands so that the redundancy is removed as much as possible. The second stage is designed to exploit the redundancy left in the band residual from the first stage. The first stage operates in two modes: interband and intraband, based on ρ(A, B). If ρ(A, B) $< \theta$, where θ is a threshold determined from experiments, intraband prediction using the median predictor defined in JPEG-LS [20] is used. Otherwise, interband prediction is used. The median predictor is also the default predictor for the first band. Given the nomenclature of Fig. 5, the estimate of pixel x is given as follows:

$$\hat{x} = \begin{cases} \min(N, W), & \text{if NW} \geq \max(\text{N, W}) \\ \max(N, W), & \text{if NW} \leq \min(\text{N, W}) \\ N + W - NW, & \text{otherwise.} \end{cases} \tag{2}$$

In the interband mode, a band is linearly predicted using the previous bands as reference (the interband prediction used will be discussed in more detail in the next section). Once the band residual is obtained from either intraband or interband prediction, the second stage, called CCAP, is applied. CCAP is described in Section II-C. The residual from the second stage is then entropy coded.

		NN	
	NW	N	NE
WW	W	X	

Fig. 5. Pixel naming for median predictor.

B. Interband Prediction

Interband prediction operates on two colocated square blocks of neighboring bands. Given blocks 1 and 2, with standard deviation of δ_1 and δ_2 and with mean of m_1 and m_2, respectively, we normalize the pixels in block 1 by

$$\bar{x} = \frac{x - m_1}{\delta_1} \tag{3}$$

where $\bar{x}$ is the normalized pixel, and x is the original pixel. If the mean and standard deviation of block 2 are available, the pixels in block 2 are predicted by

$$\hat{y} = \bar{x} * \delta_2 + m_2. \tag{4}$$

The residual is then obtained by

$$\tilde{y} = y - \hat{y}. \tag{5}$$

Both δ_2 and m_2 are required for prediction at the encoder and reconstruction at the decoder. They could be either sent to the decoder as side information or predicted using the available neighboring blocks. A large block size may reduce the overhead due to side information, while choosing a small block size may enable accurate prediction of the mean and standard deviation using the neighboring blocks.

C. Context-Based Conditional Average Prediction

There is still strong redundancy in the residual image after stage 1. The second stage, CCAP, is designed to remove the redundancy further and decrease the first-order entropy of the residuals. Note that CCAP works on the residual value of the pixel from stage 1, and therefore, the "pixel" in this section refers to the residual of the pixel obtained from stage 1. For each residual value of a pixel, the residual values of its four neighboring pixels compose a context for CCAP to derive a prediction.

1) Conditional Expectation: The optimal estimate (in the sense of mmse) of a random variable X, given a set of observation X_i, where $X_i = (x_i^1, x_i^2, \ldots, x_i^N)$, is known to be the conditional expectation of X, given X_i

$$E[x|X_i] = \sum xP\left[X = x|X_i = \left(x_i^1, x_i^2, \ldots, x_i^N\right)\right]. \tag{6}$$

For 2-D images, given a pixel $x_{i,j}$ and its context set $X_{i,j}$, the optimal estimate of the pixel is its conditional expected value $E[x_{i,j}|X_{i,j}]$. In practice, we can assume that the pixel $x_{i,j}$ is conditionally independent of pixels that are some distance from it, and hence, the conditioning variables can be limited to pixels in its causal neighborhood or causal context. For jointly Gaussian processes, the conditional expectation can be expressed as a linear combination of the observations. However, for the non-Gaussian case, the computation of the conditional expectation requires the availability of the conditional probability density function. In the case of image pixels, this is not available. Slyz and Neuhoff [21] reduced the size of the problem by replacing the conditioning variables with their vector quantized representation. In the area of text compression, Cleary and Witten [22] developed a blending approach in estimating the conditional probabilities in their development of the prediction-with-partial-match (PPM) algorithm. This approach implicitly relies on the fact that the textual information contains numerous exact repeats. As this situation is not usual in natural images, the algorithm used in PPM cannot be applied directly for image prediction. Fortunately, while we do not have exact repeats as in textual data, our objective is also not the same. We are more interested in an expected value, which can be estimated using a sample mean and which would have no meaning in the context of text compression. We apply the idea to our problem in the following manner.

Given a pixel $x_{i,j}$, a set of pixels in its causal neighborhood is defined as its causal context. The exact composition of this set depends on the scanning method (and, hence, the operational meaning of "causal") and the definition of neighborhood. For convenience, we refer to the context pixels as $C_{i,j} = (x_{i,j}^1, x_{i,j}^2, \ldots, x_{i,j}^k)$. Given a set of value $\bar{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_k)$, we say $C_{i,j} \in C_k(\bar{\alpha})$ if $x_{i,j}^r = \alpha_r$ for $r = 1, 2, \ldots, k$, i.e., the set $C_k(\bar{\alpha})$ consists of all pixels whose causal neighborhoods take on the values $\alpha_1, \alpha_2, \ldots, \alpha_k$. Then, we estimate $E[x_{i,j}|C_{i,j} = \bar{\alpha}]$ by taking the sample mean

$$\hat{\mu}_{X|\alpha} = \frac{1}{\|C_k(\bar{\alpha})\|} \sum_{(i,j):C_{i,j} \in C_k(\bar{\alpha})} x_{i,j} \tag{7}$$

where $\|C_k(\bar{\alpha})\|$ denotes the cardinality of $C_k(\bar{\alpha})$. Several issues need to be addressed before this method is put into practice. A proper size of $\|C_k(\bar{\alpha})\|$ is needed for $\hat{\mu}_{X|\alpha}$ to be a good estimate. When $\|C_k(\bar{\alpha})\|$ is not sufficiently large for $\hat{\mu}_{X|\alpha}$ to be a good estimate of $x_{i,j}$, then a neighboring pixel to $x_{i,j}$ is taken as its estimate. In addition, a good balance between the computational cost and prediction accuracy is necessary for real-time onboard compression applications that require fast context searches.

In [23], the context match is obtained by exhaustively comparing the current context with the past contexts. Given the context $C_{i,j}$ for a pixel $x_{i,j}$, we searched over the context history and declared context $C_{i,j}$ matches $C_k(\bar{\alpha})$ if

$$\left|\alpha_r - x_{i,j}^r\right| \leq T, \qquad r = 1, 2, \ldots, k \tag{8}$$

where T is an empirically determined threshold. This method turned out to work fine and provide good compression performance. However, the exhaustive search and the delay caused are somewhat prohibitive and intolerable for real-time applications.

To avoid these problems, we build a table to classify and index each context. The pixels in each context are scalar quantized, and the quantization indexes are grouped to generate a table index. Note that the context match defined in (8) does not partition the space of conditioning contexts into disjoint sets, while, in this case, the context space is partitioned into disjoint sets. The mean of the context is first removed, and we then quantize the mean-removed context pixels, the mean, and the absolute difference of the context using scalar quantization. For each pixel to be coded, a table index is obtained by concatenating all these quantization indexes. During these operations, only scalar comparison and a bit-shift operation are involved. For each context index, the table records three parameters as follows:

1) M, the total number of occurrences of the context;
2) S, the sum of pixels conditioned on this context;
3) E, the accumulated prediction error.

Having obtained the context index p for $x_{i,j}$, we examine the content of the corresponding table entry. If M is sufficiently large, the estimate of $x_{i,j}$ is obtained as follows:

$$\hat{x}_{i,j} = \frac{S - E}{M} \tag{9}$$

where $\hat{x}$ represents the average of the past pixels conditioned on the context that is specified by the context index. The accumulated prediction error E is used to adjust the estimate, i.e., for bias cancellation. If M is greater than a threshold, we consider this estimate to be valid. Otherwise, the pixel W is taken as the estimate of the pixel x. We find that five is a good threshold for a valid prediction. After this prediction, we update the table content for this context as follows:

$$\begin{cases} M \leftarrow M + 1 \\ S \leftarrow S + x \\ E \leftarrow E + \tilde{x} \end{cases} \tag{10}$$

where $\tilde{x}$ is the prediction error obtained by $\tilde{x} = x - \hat{x}$.

2) Entropy Coding of the Band Residual: The band residual from the second stage is entropy encoded. Context-based Golomb–Rice code is used for this purpose. The Golomb–Rice code is optimum for alphabets with geometric distributions and extremely efficient for computer implementation. It is used in JPEG-LS to encode residuals from a median predictor, which follows a geometric distribution. Using a Golomb–Rice code, instead of an arithmetic code, results in considerable reduction on computational complexity for a minimal sacrifice in the compression ratio. As our goal is a fast and computationally efficient compression approach, we prefer the Golomb–Rice code to arithmetic code and use it in the way similar to JPEG-LS. That is, for each pixel, the split-coding parameter for Golomb–Rice code is estimated on-the-fly based on the context of the pixel, as well as the absolute accumulated errors in this context. In addition, a run-length mode is initiated once consecutive pixels in a line are the same. This can result in a big compression saving for blocks that have many zeros.

For compression, we also used an adaptive arithmetic code (AAC) to encode the residual. An AAC does not require prior knowledge of the symbol distribution but, instead, updates the symbol distribution in a backward adaptive manner. We assume the initial distribution to be geometric. Rather than use a context-based arithmetic code, we encode each band using the AAC separately. This reduces the cost of classifying the contexts while preserving good compression performance.

III. Simulation Results

The data set for the experiment are the 1997 Airborne Visible InfraRed Imaging Spectrometer (AVIRIS)-calibrated radiance images [24]. Each image comprises several scenes, and for each scene, there are 224 bands. Each band consists of 512 × 614 pixels, and each pixel is a 16-bit signed integer. The results reported for each image are the average results of scenes. The coding of blocks follows raster order, and the block size is 64 × 64. This choice depends on several considerations: the cost to calculate the standard deviation and correlation coefficient, the efficiency of larger blocks when run-length coding is used, the memory requirement, and delay constraints.

A. Context Table

In the first stage, the switching threshold for the correlation coefficient α is set to 0.9. The mean-removed context pixels N, W, NW, NE, and mean are quantized using a 3-bit scalar quantizer, and the decision boundaries are $\{-\infty, -2^4, -2^3, -2^2, 0, 2^2, 2^3, 2^4, +\infty\}$. The sum of the absolute value of the mean-removed pixels is quantized by a 2-bit scalar quantizer with decision boundaries $\{0, 2^1, 2^2, 2^3, +\infty\}$. Therefore, the total number of bits required for indexing is 17. As the coding proceeds, the contexts that are far from the current context have gradually less effect on the prediction for the new contexts. Therefore, the weight of those remote contexts is reduced according to the procedure in (11). For each context index, we limit M to 64 and S and E up to 2 bytes. When $M > 64$, we adjust the table content as follows:

$$\begin{cases} S \leftarrow S - \frac{S}{M} + x \\ E \leftarrow E - \frac{E}{M} + \tilde{x} \\ M = 64. \end{cases} \tag{11}$$

When $S \geq 2^{16}$, we adjust S such that $S < 2^{16}$ as follows:

$$\text{while } (S \geq 2^{16})\left\{S \leftarrow S - \frac{S}{M} E \leftarrow E - \frac{E}{M} M \leftarrow M - 1\right\}.$$

This also prevents overflow if we use two-bytes unsigned short to record the accumulated sum. Therefore, for each table entry, we need five bytes, and the total memory requirement for the index table is $5 \times 2^{17} = 655$ kB.

B. Correlation Coefficient

Our simulation shows that considerable computation resource is taken by the calculation of correlation coefficient. To lower the cost for computing the correlation coefficients, we find that, if strong correlation exists, the correlation coefficient obtained using all pixels in entire blocks is almost identical with the one using only a small portion of the pixels in the blocks. Because the correlation coefficient is used for prediction only when strong correlation exists, we take only one pixel for every four pixels or take the average of the four pixels for this calculation. Our results indicate that this simplification results in considerable savings in terms of the computational cost without degrading the prediction accuracy.

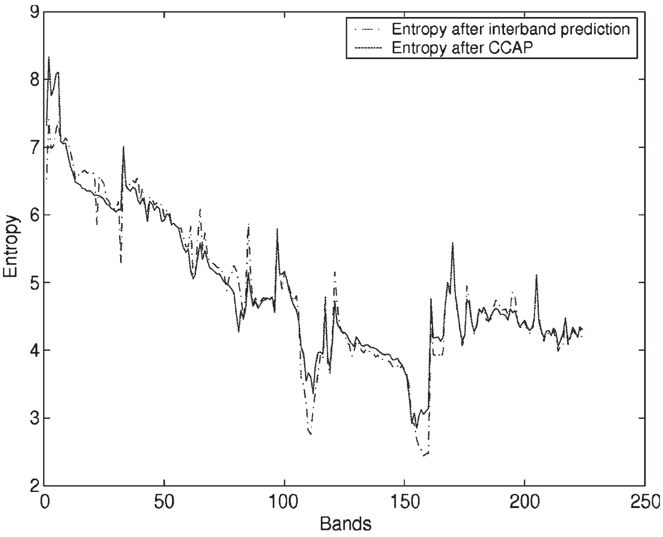

Fig. 6. Compression ratio of interband prediction versus compression ratio of residuals from CCAP.

C. Interband Prediction

As shown in [17], proper band reordering could enhance the interband prediction and, therefore, improve the compression performance. Unfortunately, the order-optimization method requires that all bands be available and exhaustive comparison be performed, which are impractical in our scenario. Thus, we develop a simplified strategy. For a block being processed, the colocated blocks in the previous four bands are used to calculate the correlation coefficient and then perform interband prediction. A block is normalized, and its correlation coefficient with four colocated blocks needs to be calculated. We can keep the normalized pixels in a 2-D matrix buffer to avoid duplicating normalization.

The interband prediction requires the mean and standard deviation of the block being predicted. Similar to pixels, the mean and standard deviation of a block are very likely to be highly correlated with those of the blocks that are colocated in previous bands. Taking advantage of available blocks and their correlation with the colocated blocks in previous bands, the mean and standard deviation of the current block was differentially coded (except for the first block at the top-left corner) and provided to the decoder.

D. Compression Results

Fig. 6 shows the compression ratios from the first and the second stages. As shown, the CCAP significantly improves

TABLE I
COMPARISON OF COMPRESSION RATIOS WITH OTHER SCHEMES

AVIRIS(1997)	JPEG-LS	JPEG-LS Differential	LPVQ	SLSQ	SLSQ(OPT)	M-CALIC	CCAP (AAC)	CCAP (GR)	LUT
Cuprite	2.09	2.91	3.18	3.15	3.24	3.21	3.25	3.16	3.44
Jasper Ridge	2.00	2.81	2.88	3.15	3.23	3.16	3.23	3.15	3.23
Low Altitude	2.14	2.70	2.94	2.98	3.04		3.03	2.94	
Lunar lake	1.99	2.93	3.28	3.15	3.27	3.23	3.22	3.17	3.40
Average	2.06	2.83	3.07	3.11	3.19	3.16	3.18	3.07	3.36

the compression performance for most bands. However, for noiselike bands and highly uncorrelated band residuals, it may not achieve compression and maybe, in fact, increases the data rate. Thus, we use CCAP only if it improves the compression ratio and send a decision bit to indicate the selection to the decoder.

Table I shows the compression ratios of the proposed approach, as compared with other schemes [11], [14]. We can see that the proposed approach outperforms both JPEG-LS and JPEG-LS differential schemes, in which JPEG-LS compresses the difference between two adjacent bands. This is not surprising, as these are general-purpose image-compression algorithms. We note this in order to emphasize the importance of developing application-specific compression techniques. In addition, the proposed scheme CCAP with AAC outperforms both LPVQ and SLSQ by about 34% and 4%, respectively, and is comparable to the optimized SLSQ and M-CALIC. The proposed CCAP with Golomb–Rice code is comparable to LPVQ and comparable also to others. Note that this compression performance is obtained by using the simple Golomb–Rice code rather than the context-based arithmetic code used by other algorithms.

E. Complexity

In order to evaluate the time complexity and compare it with other methods, we may divide a coding procedure into two stages, prediction and entropy coding, and consider the complexity of each stage separately. The prediction stage consists of correlation-coefficient calculation, interband prediction, and CCAP. The calculation of the correlation coefficient is simplified by taking a small portion of the pixels of each block. Most methods using the correlation coefficient as a switch, including 3-D CALIC and M-CALIC, do not sufficiently simplify this operation. The interband prediction requires only one addition and one multiplication for each pixel. For each pixel in CCAP, seven additions are needed to remove the mean and obtain the absolute difference, and two additions and two divisions are needed to obtain the conditional average and bias cancellation. Therefore, we need a total of ten additions, two divisions, and one multiplication for each pixel in the interband-prediction mode. This is roughly equivalent to SLSQ, which requires four multiplications and six additions for each pixel prediction for the interband predictor. It is lower in complexity than 3-D CALIC and M-CALIC. However, it has a higher complexity than the LUT technique, in which the prediction of one pixel requires only two memory operations: a fetch operation and store operation.

For entropy coding, the Golomb–Rice code is well known to be considerably less complex than arithmetic coding, which is employed in LPVQ, 3-D CALIC, M-CALIC, and SLSQ. For LUT, the complexity of adaptive-range coding is around half of arithmetic coding [26]. According to [10], the complexity of arithmetic coding accounts for around 70% of the computation time of 2-D CALIC. Approximately, 3-D CALIC doubles the computation time of 2-D CALIC. It is reasonable to assume that the additional costs are mainly from the interband prediction and context modeling. By looking at these statistics, we can assume that arithmetic coding roughly accounts for 35% of the computation time of 3-D CALIC and M-CALIC (M-CALIC only has around 10% complexity increase over 3-D CALIC). However, in our method, entropy coding by Golomb–Rice codes takes less than 10% of the total compression time.

We compared the complexity of the proposed algorithm with JPEG-LS [25] on a personal computer with a Pentium IV, 2.8-GHz processor, and Windows XP system. The compression time was recorded using the clock() function. We compressed the AVIRIS image set and took the average running time as the reference. The average time (in seconds per band) is 0.05 for JPEG-LS and 0.31 for our method. That is, our unoptimized code is around six times slower than JPEG-LS. We do not have other compression methods available. However, based on the relative compression time with JPEG-LS presented in [10], our method is around 35% faster than 3-D CALIC and 42% faster than M-CALIC. Although the comparison might be biased by code implementation, optimization, hard-drive access, and memory access, it may provide some insight on the complexity. We do not have running-time values for the LUT method. Considering that the LUT method is inherently simple, we expect that its complexity will be low. However, no definitive conclusion can be drawn.

IV. CONCLUSION

We have proposed a novel prediction method based on the context history of hyperspectral images. A context index table is built up for context recording, fast-context search, and context match. The approach switches between interband and intraband modes based on the correlation coefficient. The residual from interband prediction and intraband prediction is further processed using the CCAP method and then coded by Golomb–Rice coding. We apply the proposed method to hyperspectral-image compression and obtain competitive compression ratios. The advantage of the method is its low complexity and computational cost, which make it promising for onboard applications.

REFERENCES

[1] M. J. Ryan and J. F. Arnold, "The lossless compression of AVIRIS images by vector quantization," *IEEE Trans. Geosci. Remote Sens.*, vol. 35, no. 3, pp. 546–550, May 1997.

[2] M. R. Pickering and M. J. Ryan, "Efficient spatial-spectral compression of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1536–1539, Jul. 2001.

[3] G. Motta, F. Rizzo, and J. Storer, "Partitioned vector quantization: Application to lossless compression of hyperspectral images," in *Proc. ICASSP*, Apr. 6–10, 2003, vol. 3, pp. III-241–III-244.

[4] G. Motta, F. Rizzo, and J. A. Storer, "Compression of hyperspectral imagery," in *Proc. DCC*, Mar. 2003, pp. 333–342.

[5] *Lossless data compression, CCSDS report concerning space data systems standards*, 1997, Washington DC: Green Book. 120.0-G-1.

[6] N. D. Memon, K. Sayood, and S. Magliveras, "Lossless compression of multispectral image data," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 2, pp. 282–289, Mar. 1994.

[7] R. E. Roger and M. C. Cavenor, "Lossless compression of AVIRIS images," *IEEE Trans. Image Process.*, vol. 5, no. 5, pp. 713–719, May 1996.

[8] B. Aiazzi, P. Alba, L. Alparone, and S. Baronti, "Lossless compression of multi/hyper-spectral imagery based on a 3-D fuzzy prediction," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 5, pp. 2287–2294, Sep. 1999.

[9] X. Wu and N. D. Memon, "Context-based lossless interband compression-extending CALIC," *IEEE Trans. Image Process.*, vol. 9, no. 6, pp. 994–1001, Jun. 2000.

[10] E. Magli, G. Olmo, and E. Quacchio, "Optimized onboard lossless and near-lossless compression of hyperspectral data using CALIC," *IEEE Geosci. Remote Sens. Lett.*, vol. 1, no. 1, pp. 21–25, Jan. 2004.

[11] F. Rizzo, B. Carpentieri, G. Motta, and J. A. Storer, "High performance compression of hyperspectral imagery with reduced search complexity in the compressed domain," in *Proc. DCC*, Mar. 23–25, 2004, pp. 479–488.

[12] M. Slyz and L. Zhang, "A block-based inter-band lossless hyperspectral image compressor," in *Proc. DCC*, Mar. 29–31, 2005, pp. 427–436.

[13] F. Rizzo, B. Carpentieri, G. Motta, and J. A. Storer, "Low-complexity lossless compression of hyperspectral imagery via linear prediction," *IEEE Signal Process. Lett.*, vol. 12, no. 2, pp. 138–141, Feb. 2005.

[14] J. Mielikainen and P. Toivanen, "Clustered DPCM for the lossless compression of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 12, pp. 2943–2946, Dec. 2003.

[15] J. Mielikainen, "Lossless compression of hyperspectral images using lookup tables," *IEEE Signal Process. Lett.*, vol. 13, no. 3, pp. 157–160, Mar. 2006.

[16] B. Penna, T. Tillo, E. Magli, and G. Olmo, "Progressive 3-D coding of hyperspectral images based on JPEG 2000," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 125–129, Jan. 2006.

[17] S. Tate, "Band ordering in lossless compression of multispectral images," in *Proc. DCC*, Mar. 1994, pp. 311–320.

[18] J. Zhang and G. Liu, "An efficient reordering prediction-based lossless compression algorithm for hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 2, pp. 283–287, Apr. 2007.

[19] X. Wu and N. D. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.

[20] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[21] M. J. Slyz and D. L. Neuhoff, "A nonlinear VQ-based predictive lossless image coder," in *Proc. DCC*, Mar. 1994, pp. 304–310.

[22] J. G. Cleary and I. H. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Trans. Commun.*, vol. COM-32, no. 4, pp. 396–402, Apr. 1984.

[23] H. Wang, S. D. Babacan, and K. Sayood, "Lossless hyperspectral image compression using context-based conditional averages," in *Proc. DCC*, Mar. 2005, pp. 418–426.

[24] *AVIRIS Images*, Jet Propulsion Laboratory, NASA. [Online]. Available: http://aviris.jpl.nasa.gov/html/aviris.overview.html

[25] *HP Labs LOCO-I/JPEG-LS Home Page*. [Online]. Available: http://www.hpl.hp.com/loco/index.htm

[26] Michael Schindler, "Range encoder Homepage," Oct. 1999. [Online]. Available: http://www.compressconsult.com/rangecoder/

Hongqiang Wang was born in Baoji, China, in 1975. He received the B.S. degree in electronics and information systems from Peking University, Beijing, China, in 1998 and the M.Eng. degree in coding, modulation, and signal processing from Nanyang Technological University, Singapore, in 2003. He is currently working toward the Ph.D. degree in electrical engineering in the Department of Electrical Engineering, University of Nebraska–Lincoln.

From 1998 to 2000, he was a Design Engineer with ZTE Corporation, Guangdong, China. He was a Consultant with Panasonic Singapore Laboratories, Singapore, from 2002 to 2003, and with Reuters, Hong Kong, in 2003. His research interests include image, video, and audio coding.

S. Derin Babacan was born in Istanbul, Turkey, in 1981. He received the B.Sc. degree from Boðaziçi University, Istanbul, in 2004 and the M.Sc. degree from Northwestern University, Evanston, IL, in 2006, where he is currently working toward the Ph.D. degree in electrical engineering in the Department of Electrical Engineering.

He is a Research Assistant with the Image and Video Processing Laboratory, Northwestern University. His primary research interests include image and video compression, restoration, superresolution, and computer vision.

Khalid Sayood (S'78–M'79–SM'95) received the B.S. and M.S. degrees in electrical engineering from the University of Rochester, Rochester, NY, in 1977 and 1979, respectively, and the Ph.D. degree in electrical engineering from Texas A&M University, College Station, in 1982.

From 1995 to 1996, he served as the Founding Head of the Computer Vision and Image Processing Group at the Turkish National Research Council Informatics Institute. From 1996 to 1997, he was a Visiting Professor with Boðaziçi University, Istanbul, Turkey. Since 1982, he has been with the University of Nebraska–Lincoln, where he is currently serving as the Henson Professor of Engineering in the Department of Electrical Engineering. He is the Author of *Introduction to Data Compression, 3rd ed.* (Morgan Kaufmann, 2005), *Understanding Circuits: Learning Problem Solving Using Circuit Analysis* (Morgan Claypool, 2005), and *Learning Programming Using MATLAB* (Morgan Claypool, 2006) and the Editor of *Lossless Compression Handbook* (Academic Press, 2002). His research interests include bioinformatics, data compression, and joint-source channel coding.