

Reduced Complexity Wavelet-Based Predictive Coding of Hyperspectral Images for FPGA Implementation *

Agnieszka C. Miguel[#] Amanda R. Askew[†] Alexander Chang[†]
Scott Hauck[‡] Richard E. Ladner[†]
Eve A. Riskin[‡]

[#] Department of Electrical and Computer Engineering,
Seattle University, 900 Broadway, Seattle, WA 98122-4340

[†] Department of Computer Science and Engineering, Box 352350,
University of Washington, Seattle, WA 98195-2350

[‡] Department of Electrical Engineering, Box 352500,
University of Washington, Seattle, WA 98195-2500

Abstract

We present an algorithm for lossy compression of hyperspectral images for implementation on field programmable gate arrays (FPGA). To greatly reduce the bit rate required to code images, we use linear prediction between the bands to exploit the large amount of inter-band correlation. The prediction residual is compressed using the Set Partitioning in Hierarchical Trees algorithm. To reduce the complexity of the predictive encoder, we propose a bit plane-synchronized closed loop predictor that does not require full decompression of a previous band at the encoder. The new technique achieves similar compression ratios to that of standard closed loop predictive coding and has a simpler on-board implementation.

1 Introduction

Every day, NASA collects and stores large amounts of hyperspectral data. For example, one Moderate Resolution Imaging Spectroradiometer (MODIS) alone can produce hyperspectral data that require up to 225 Gbytes of storage per day. The Terra spacecraft produces 194 Gbytes of data per day [1]. The hyperspectral images are transmitted to the ground station,

*This work appeared in part in the Proceedings of the NASA Earth Science Technology Conference, 2003, and in the Proceedings of the Data Compression Conference, 2004. Research supported by NASA Contract NAS5-00213 and National Science Foundation grant number CCR-0104800. Scott Hauck was supported in part by an NSF CAREER Award and an Alfred P. Sloan Research Fellowship. Contact information: Professor Eve Riskin, University of Washington, Box 352500, Seattle, WA 98195-2500, (206) 685-2313, riskin@ee.washington.edu.

stored, and used to identify occurrences of fire, water, and snow on the Earth.

The huge amount of data generated by satellites presents a compression challenge. In this research, we code the hyperspectral data with the Set Partitioning in Hierarchical Trees (SPIHT) algorithm [2], which is a wavelet-based technique that codes images with both high compression ratios and high fidelity. SPIHT was originally designed as a sequential algorithm; however, with some modifications, it can be parallelized for implementation on field programmable gate arrays (FPGAs) [3] and therefore has great potential for applications where the compression is performed in hardware on the satellite.

To reduce the bit rate required to code hyperspectral images, we use linear prediction between the bands. Each band, except the first one, is predicted by previously transmitted band. Once the prediction is formed, it is subtracted from the original band, and the residual (difference image) is compressed using SPIHT. Because different bands are used for different purposes, we compress all bands to the same fidelity.

To compute the exact difference between a band and its prediction, the encoder must have access to the decoded version of the band used for prediction; however, such a closed loop system requires a full implementation of the decoder on the satellite, which increases the complexity of on-board applications. In this article we present a new prediction technique, *bit plane-synchronized closed loop prediction*, that significantly reduces the complexity of the encoder [4]. Instead of requiring the encoder to fully reconstruct the compressed band from which the current band is predicted, the encoder and the decoder simply use the same integral number of full bit planes of the wavelet-coded difference image of the band used for prediction. This enables the encoder to be less complex because, while it must still do an inverse wavelet transform, full decompression on-board the satellite is avoided. The proposed prediction method is very promising in that for the same target fidelity, the average bit rate is only slightly higher than for traditional predictive coding.

The paper is organized as follows. In Section 2, we review related background material. In Section 3, we describe algorithms for predictive coding of hyperspectral images. In Section 4, we introduce our new reduced complexity encoder. We present results of our algorithm on hyperspectral images in Section 5, and conclude in Section 6.

2 Background

In this section, we present related work. We first review satellite images, the SPIHT algorithm, and predictive coding. Then, we discuss prior work in hyperspectral image compression and finish with a discussion on FPGAs.

2.1 Satellite Images

NASA has launched numerous Earth-imaging satellites in the past decade, which take pictures of the Earth to help understand our planet's topography, weather, and environment. Many of these instruments take multiple pictures of the same location, but at different wave-

lengths, to provide a greater range of observations. Such images can involve tens (“multi-spectral”) or hundreds (“hyperspectral”) of individual spectral bands. As such, a hyperspectral image can be viewed as a 3D data cube, with the X and Y dimensions representing different coordinates on the Earth’s surface, while the third dimension is the band, representing the frequency being imaged. The actual data values are the intensity of the light at that wavelength from that location on the Earth. For example, a MODIS image might be 2030x1354 spatially, with 36 spectral bands holding 16-bit unsigned pixel values. An AVIRIS image might be 512x512 spatially, with 224 spectral bands, and a 16-bit pixel value.

2.2 Set Partitioning in Hierarchical Trees

SPIHT is a progressive image coder, which first approximates an image with a few bits of data, and then improves the quality of approximation as more information is encoded. As shown in Figure 1, the encoder first performs a wavelet transform on the image pixels. Then, the wavelet coefficients are encoded one bit plane at a time. The embedded bit stream, in which the later bits refine the earlier bits, can be truncated at any time (see Figure 2). Bit plane encoding and decoding take significantly more time than the wavelet transform.

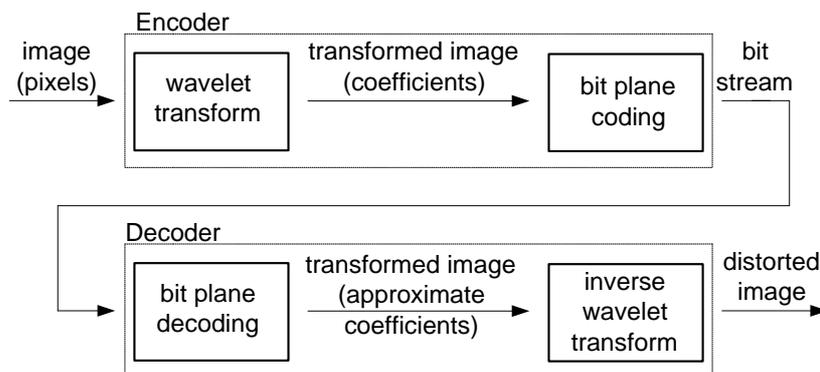


Figure 1: Block diagram of SPIHT.

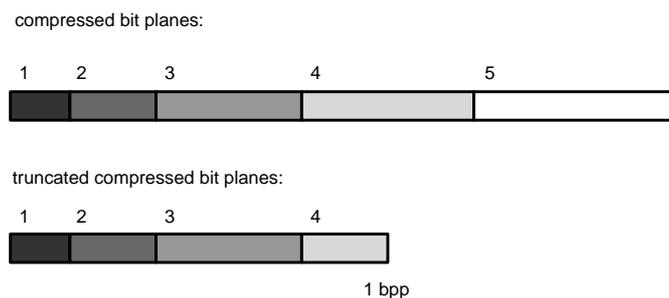


Figure 2: Bit plane coding.

2.3 Predictive Coding

Predictive coding has been a popular data compression technique for years. Prediction exploits the correlation in spatial or temporal dimensions, or context, to improve compression performance over independent coding. It is used in both lossy and lossless coding. For example, differential pulse code modulation (DPCM) [5] uses prediction to improve performance over standard PCM. The MPEG video coding standard [6] uses temporal prediction to significantly improve compression ratios over independent coding of video frames. Predictive vector quantization (VQ) [7, 8, 9, 10] exploits spatial correlation over a larger region of an input image or speech signal to give improvements over memoryless VQ [11, 9]. Usually in predictive VQ, the design of the predictor is open-loop for simplicity (the predictor is optimized using unquantized samples and then fixed), although it is expected that a somewhat higher PSNR would be obtained by using a closed-loop design (the predictor and quantizer are jointly optimized) [8].

2.4 Previous Work in Hyperspectral Image Compression

The proposed techniques for lossy compression of hyperspectral images can be classified into two types: vector quantization [12, 13, 14] and transform-based [15, 16, 17] algorithms. Motta, Rizzo, and Storer [12] designed a product VQ with an algorithm to determine how to form subvectors across bands. In [18], Rizzo, Carpentieri, Motta, and Storer modify this method to include a low-complexity encoder. Qian et al. [13] generated separate sub-codebooks for regions with similar spectral characteristics. Ryan and Pickering [14] used mean-normalized VQ followed by the discrete-cosine transform (DCT) in the spatial and spectral domains and entropy coding.

Markas and Reif [19] applied the DCT or the Karhunen-Loeve (KL) transform in the spectral domain and the discrete wavelet transform in the spatial domain, followed by uniform scalar quantization and block-based encoding using multidimensional bitmap trees. Abousleman, Marcellin, and Hunt in [20] proposed using DPCM for spectral decorrelation and 2-D DCT for spatial decorrelation combined with entropy-constrained trellis coded quantization (ECTCQ). Lee, Younan, and King [16] used different 1-D transforms to obtain spectral decorrelation (KL, DCT, and the difference pulse-coded modulation (DPCM)) and applied JPEG 2000 to the resulting data.

Tang, Cho, and Pearlman compared the performance of several 3-D versions of SPIHT on hyperspectral data with the performance of JPEG2000 [17]. The algorithms included the original 3D-SPIHT, 3D-SPIHT with asymmetric trees (AT-3DSPIHT), and the 3D Set Partitioned Embedded Block method. AT-3DSPIHT outperformed the other algorithms by 0.2-0.9 dB. All of the SPIHT-based algorithms were significantly better than JPEG2000 applied to each band separately. Dragotti, Poggi, and Ragozini [21] modified the 3D-SPIHT algorithm to better compress multispectral images. In the first method, they performed a 3D transform which consisted of the wavelet transform in the spatial domain and the KL transform in the spectral domain. The 3D transform was followed by 3D-SPIHT coding. In the second method, 2D wavelet transform was first taken in the spatial domain. Then,

spectral vectors of pixels were vector quantized and gain-driven SPIHT was used.

Linear prediction as a method to reduce inter-band correlation was investigated by Memon in [22] who proposed adaptive reordering of the spectral components of each pixel followed by a piecewise linear function at a specified error tolerance. Rao and Bhargava [23] used simple block-based linear inter-band prediction followed by a block-based DCT. To take advantage of linear prediction between bands, Tate in [24] explored unconstrained optimal reordering of the multispectral bands followed by linear prediction, which uses spatial neighborhoods to predict each pixel and arithmetic coding.

An appropriate distortion measure for compressed hyperspectral data was investigated by Ryan and Arnold in [25]. Their goal was to find a measure that is a suitable error metric for decompressed data used in various scientific algorithms instead of viewed by humans. The proposed distortion measure was the Percentage Maximum Absolute Distortion which guarantees that each pixel in the reconstructed image is within a maximum percentage distance of its original value.

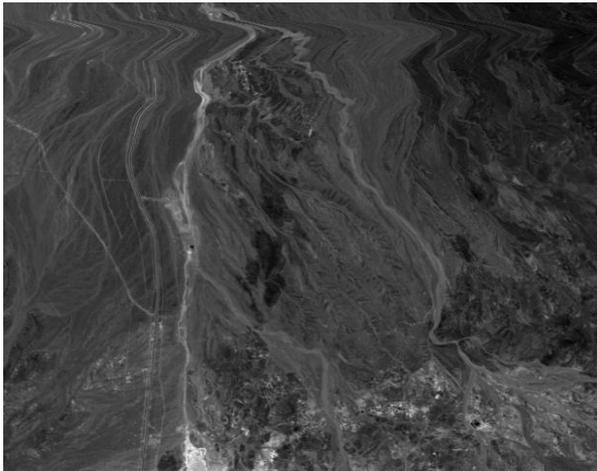
In addition to lossy compression, lossless hyperspectral image coding has been also widely investigated. See [24, 26, 27, 28, 29, 30, 31, 32] for more detail.

2.5 FPGAs

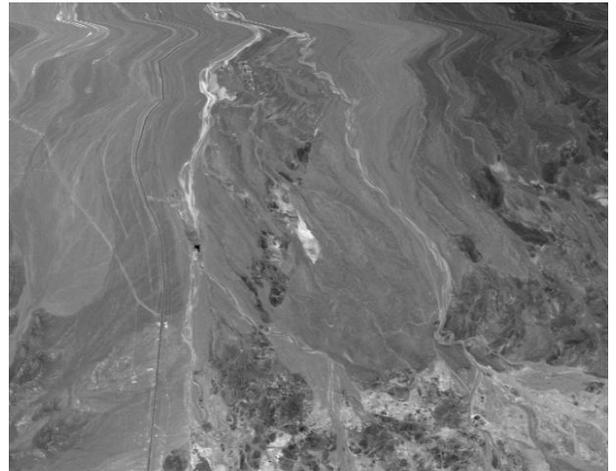
Field-Programmable Gate Arrays are chips that can be programmed and reprogrammed to implement complex digital logic [33]. Current chips may run at speeds of 100MHz or more, with logic capacities in the millions of gates. As such they can implement large computation chunks, or even complete systems. Because of their reprogrammability, they are particularly attractive in remote applications, where configurations can be changed and upgraded by simply shipping a setup file. The chips are programmed by a binary file, similar to a program executable, which sets configuration points in the chip to program in the desired circuit. While these chips are malleable like software, they implement hardware circuits, and have similar capabilities and requirements to custom logic devices. For our application, satellite image compression, their mix of high performance, high gate count, and remote reconfigurability make them particularly attractive.

3 Predictive Coding of Hyperspectral Images Using SPIHT

Because each band of a hyperspectral image corresponds to the same location on Earth, there is a high level of correlation between the bands (see Figure 3 for an example). However, bands corresponding to different wavelengths have different dynamic ranges, and thus, do not lend themselves easily to simple difference coding. For example, in Figure 3, Bands 30 and 200 of a 224-band image of the Cuprite geology site are highly correlated, yet a simple difference between the two would contain significant energy. It has been suggested in the literature that the pixel values across different bands are linearly related [24, 34] and simple linear prediction has been successfully used to remove the spectral correlation between



(a) Band 30



(b) Band 200

Figure 3: Sample bands 30 and 200 of a 224-band image of the Cuprite geology site.

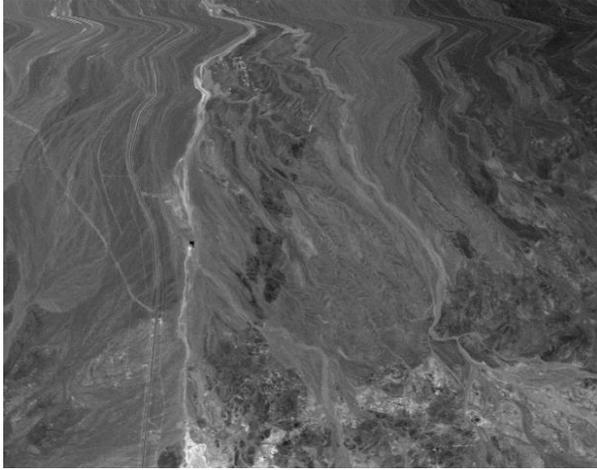
bands [20].

We also use linear prediction to take advantage of correlation between bands. Assume there are m bands B_i for $0 \leq i < m$. We define an additional *root band* B_m in which each pixel is the constant 1. This band will be the only band that is not predicted by another band. Each band B_i ($0 \leq i < m$) can be linearly predicted from another band B_j ($0 \leq j < m$) and the root band as described in Equations (1). The values a_{ij} and c_{ij} are the *prediction coefficients* and P_{ij} is the prediction of the current band B_i from a previously transmitted band B_j . The difference D_{ij} between B_i and P_{ij} is a residual and can usually be compressed well. Once D_{ij} is transmitted, band B_i can be recovered by adding D_{ij} to the prediction P_{ij} .

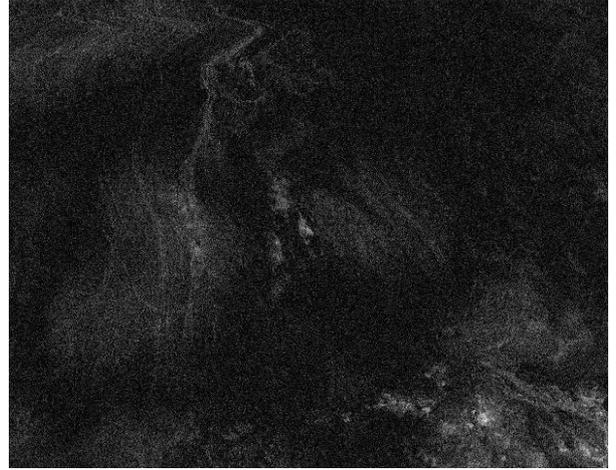
$$\begin{aligned}
 P_{ij} &= a_{ij}B_j + c_{ij}B_m \\
 D_{ij} &= B_i - P_{ij} \\
 B_i &= P_{ij} + D_{ij}.
 \end{aligned}
 \tag{1}$$

Figure 4 visually shows the advantage of linear prediction over direct difference coding. Figure 4 (a) is the simple difference of Bands 77 and Band 78 ($B_{77} - B_{78}$), whereas Figure 4 (b) is the result of solving Equation 1 when predicting Band 77 from Band 78. Notice that the difference corresponding to the linearly predicted band (right) contains a great deal less energy than the simple difference (left).

Note that the prediction P_{im} requires that $a_{im} = 0$ so that the prediction only depends on the value of c_{ij} . We assume that the prediction coefficients are known to both the encoder and decoder by some prior communication. The quality of a particular prediction can be measured by its *prediction mean squared error (PMSE)*, $\|D_{ij}\|^2/n$, where n is the number of pixels in a single band. Generally, the larger the PMSE, the more bits are needed to



(a) Difference band $B_{77} - B_{78}$.



(b) Difference band 77 when band 77 is predicted from band 78 using linear prediction ($D_{77\ 78}$).

Figure 4: Band 77 - Band 78 (left) and the difference band when band 77 is predicted from band 78 (right).

compress the difference. The PMSE depends on a good choice of a_{ij} and c_{ij} . If $0 \leq i, j < m$, $i \neq j$, then a natural choice for a_{ij} and c_{ij} are values that minimize the PMSE. These can be calculated by least squares fit [35]. The value c_{im} that minimizes the PMSE $\|D_{im}\|^2/n$ is simply the average pixel value of the band B_i .

A *band prediction ordering* is a function $\sigma : \{0, \dots, m-1\} \rightarrow \{0, \dots, m\}$. That is, except for band B_m , band B_i is predicted by band $B_{\sigma(i)}$. The function σ must satisfy the following property: For each i such that $0 \leq i < m$, there is a sequence $i = i_1, i_2, \dots, i_k = m$ such that $i_{j+1} = \sigma(i_j)$ for $1 \leq j < k$. An alternative definition is that a prediction order is a tree with nodes labeled uniquely from $\{0, 1, \dots, m\}$ with root labeled m . For $0 \leq i < m$, i 's parent in the tree is $\sigma(i)$.

We measure the quality of the prediction ordering σ as the *average PMSE*:

$$\frac{1}{mn} \sum_{i=0}^{m-1} \|D_{i,\sigma(i)}\|^2. \quad (2)$$

3.1 Basic Band Prediction Orderings

The simplest band ordering is the *forward monotonic ordering* where $\sigma(0) = m$ and $\sigma(i) = i - 1$ for $1 \leq i < m$, and the *reverse monotonic ordering* where $\sigma(i) = i + 1$ for $0 \leq i < m$. There are two relatively easy-to-compute alternatives that are significantly better than the monotonic orderings. These alternatives are based on examining the $m \times (m + 1)$ *prediction matrix*, where the (i, j) -th entry is $\|D_{ij}\|^2/n$, the PMSE.

Figure 5 is an example of a prediction matrix. The horizontal axis represents the predictor band numbers and the vertical axis represents the predicted band numbers. The darker

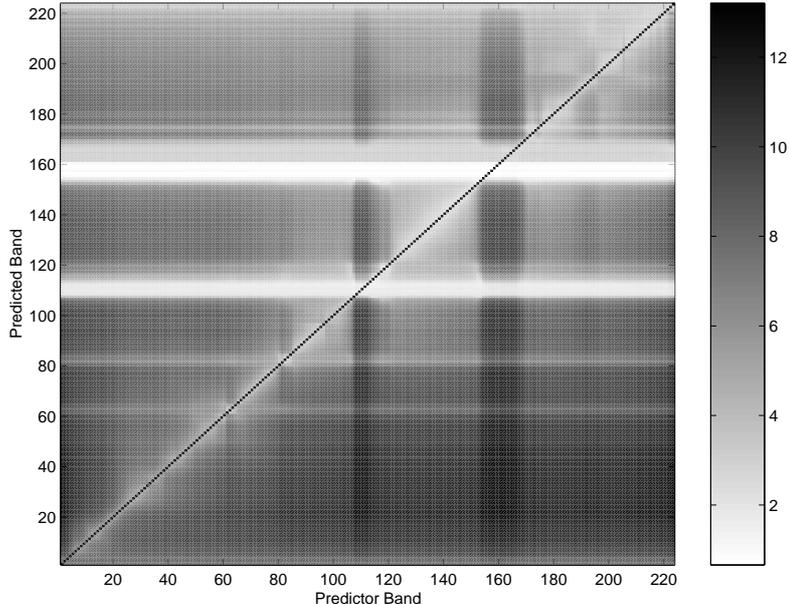


Figure 5: Prediction matrix (log scale).

color represents larger values, which is where the prediction does not perform well. Clearly, some bands do not perform well as predictors, while other bands are very easily predicted. For example, bands 110 and 160 do not predict others well, whereas bands 110 and 158 are well predicted by any other band.

To take advantage of the fact that some bands are better predictors than others we define the *best forward ordering* by choosing $\sigma(i) < i$ or $\sigma(i) = m$ that minimizes $\|D_{i,\sigma(i)}\|^2/n$ for $0 \leq i < m$. That is, the bands are predicted smallest to largest, and a particular band is predicted by the best band with a smaller number, with the exception of the root band. Similarly, we can define the *best reverse ordering* by choosing $\sigma(i) > i$ that minimizes $\|D_{i,\sigma(i)}\|^2/n$ for $1 \leq i < m$. That is, the bands are predicted largest to smallest, and a particular band is predicted by the best band with a smaller number. Both best orderings can be computed in $O(m^2)$ time once the prediction matrix is constructed.

We also consider the *optimal ordering* in which there is no limit on which band can predict other bands. We formulate the problem of determining the best ordering as a graph problem — more specifically, the problem of finding the minimum weight rooted spanning tree on a directed graph. The directed graph has $m + 1$ vertices representing the m bands and the root band. The root band is the root of the spanning tree. The directed graph has a directed edge from j to i if, $i \neq j$, $0 \leq i < m$ and $0 \leq j \leq m$. The weight of the edge from j to i is $\|D_{ij}\|^2/n$, a value in the prediction matrix. A minimum spanning tree is a spanning tree that minimizes the sum of all the weights of the edges in the spanning tree. A minimum spanning tree T defines a prediction ordering σ_T as followings. If (j, i) is a directed edge in T , then $\sigma_T(i) = j$. That is, band B_i is predicted from band B_j if (j, i) is an edge in the spanning tree. The fact that T has a minimum sum of weights ensures that Equation (2) is minimized.

The algorithm for finding the minimum spanning tree in a directed graph was first developed in the 1960s [36]. It was then applied in the 1970s [37] to solve network flow problems and its implementation was further improved in [38]. The best complexity bound for finding the minimum spanning tree in a directed graph is $O(n^2)$ [37, 38], but these algorithms are significantly more complex than computing the other orderings. The minimum spanning tree technique was also used by S.R Tate [24] to find an optimal band ordering for lossless compression of multispectral images and by Pavel Kopylov and Pasi Fränti [39] to find an optimal layer ordering for lossless compression of multilayered maps. In both these works the weight on the edge from j to i was the size of the losslessly compressed difference D_{ij} . We use the PMSE because it approximately represents the size of the lossy compressed difference regardless of the amount of loss required.

Examples of the five possible orderings are shown in Fig. 6. Table 1 lists the average PMSE over the 224-band Cuprite image for all of these orderings. As can be seen, the best reverse ordering is actually very close to the optimal ordering (within 0.41%). Since the running time of the best reverse ordering is much faster than the optimal, we use it for all simulations in this paper.

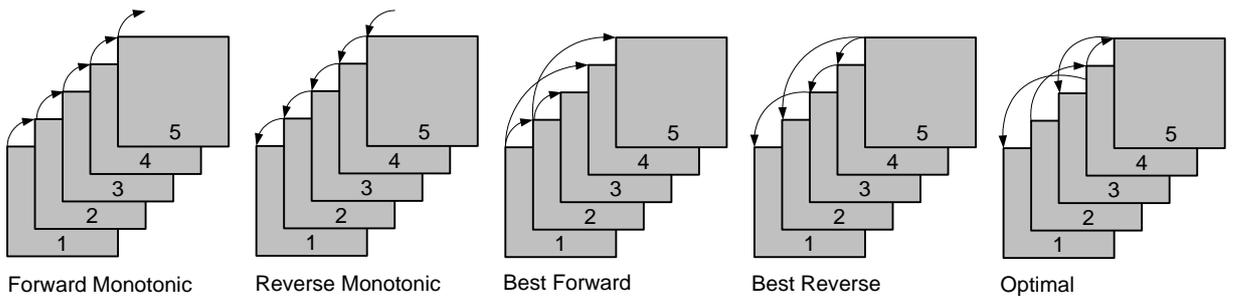


Figure 6: Prediction band ordering.

Table 1: Average PMSE for the Cuprite Image Set for 5 Different Band Orderings.

Ordering Type	Average PMSE
Forward Monotonic	254.48
Best Forward	252.61
Reverse Monotonic	183.65
Best Reverse	175.24
Optimal	174.52

3.2 Target MSE

For each band i , the quality of a particular compression scheme can be measured by its *compression mean square error (MSE)*, $\|B_i - \hat{B}_i\|^2/n$, where n is the number of pixels in a single band and \hat{B}_i is the decoded band i . In this research, we encode each band to the same compression MSE, which we call the *target MSE*, and compare the resulting bit rates. The

SPIHT algorithm can be modified to keep track of the MSE of the wavelet coefficients which approximates the MSE of the residuals and original bands. For this study we used binary search to find the bit rate for given target MSE.

Prediction significantly improves the compression ratio. For example, as shown in Figure 7, for the Cuprite image (614x512, 224 bands, 16-bit integer data), when all of the bands are encoded to a target MSE of 100 per band, using prediction increases the compression ratio from 8:1 to 33:1.

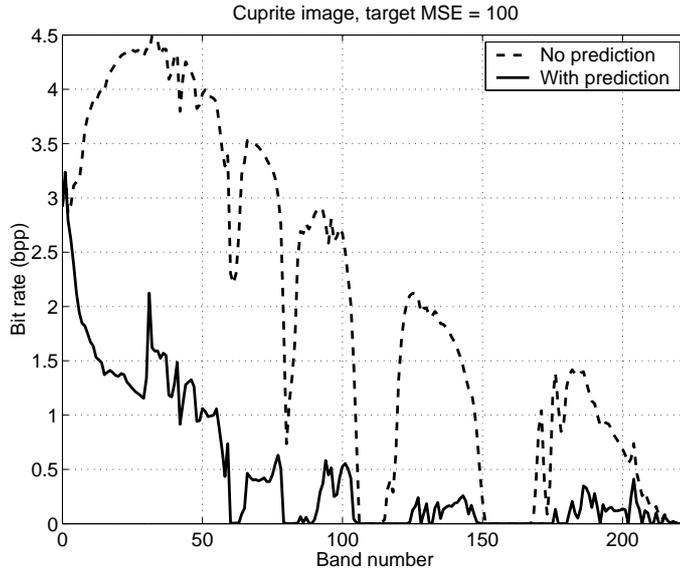


Figure 7: Comparison of bit rates required to code the Cuprite image to MSE=100 with and without prediction for best reverse ordering.

3.3 Standard Closed Loop Prediction

To predict the current band, a previous band is needed. In *closed loop prediction*, shown in Figure 8, the decompressed version of a previously encoded band is used for prediction by both the transmitter and receiver.

Let σ be a prediction ordering. As described in Equations (3), the transmitter uses a decompressed previous band $\hat{B}_{\sigma(i)}$ to form $P_{i,\sigma(i)}$, the prediction of original band B_i . Next, $P_{i,\sigma(i)}$ is subtracted from B_i to obtain the difference $D_{i,\sigma(i)}$, which is then coded with SPIHT to the bit rate which yields the target MSE. The decompressed difference band $\hat{D}_{i,\sigma(i)}$ is summed with $P_{i,\sigma(i)}$ to obtain \hat{B}_i . Finally, \hat{B}_i is stored in the encoder and decoder so that it can be used to predict some other band, if necessary. Note that this method requires the transmitter to implement the decoder, which increases computational complexity.

$$\begin{aligned}
P_{i,\sigma(i)} &= a_{i,\sigma(i)}\hat{B}_{\sigma(i)} + c_{i,\sigma(i)}B_m \\
D_{i,\sigma(i)} &= B_i - P_{i,\sigma(i)} \\
\hat{B}_i &= P_{i,\sigma(i)} + \hat{D}_{i,\sigma(i)}
\end{aligned} \tag{3}$$

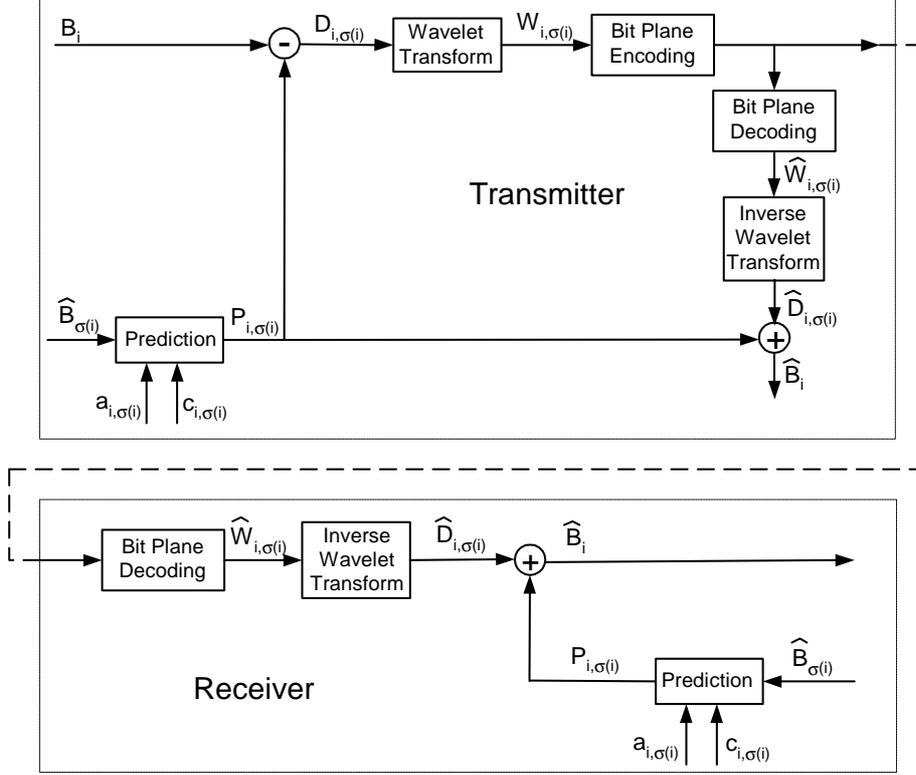


Figure 8: Standard closed loop prediction.

4 Bit Plane-Synchronized Closed Loop Prediction

As a lower complexity solution for on-board implementation, we introduce a new kind of predictive coder, the *bit plane-synchronized closed loop predictor*. We take advantage of the fact that the SPIHT algorithm can be split into two steps: wavelet transform and bit plane coding. We also exploit the fact that the wavelet transform step requires much less computation than the bit plane encoding step. To eliminate the bit plane decoding step from the transmitter, we will predict using only full bit planes of the wavelet transform.

4.1 The Algorithm

The transmitter first performs the wavelet transform on the difference band $D_{i,\sigma(i)}$ to obtain $W_{i,\sigma(i)}$. Let $R(W_{i,\sigma(i)})$ be the bit rate required to encode $W_{i,\sigma(i)}$ to the target MSE. This corresponds to stopping the encoder mid-bit-plane, for example, in bit plane number $k + 1$.

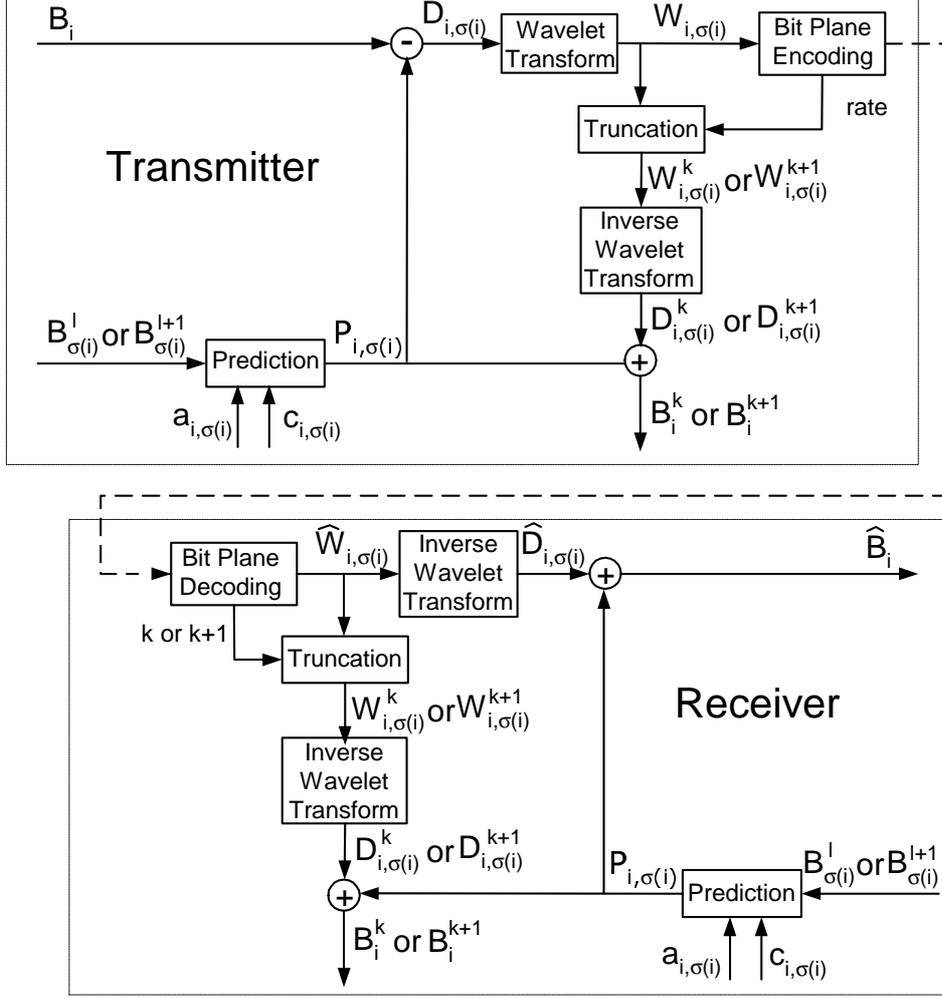


Figure 9: Bit plane-synchronized closed loop prediction.

Let $W_{i,\sigma(i)}^k$ and $W_{i,\sigma(i)}^{k+1}$ be the wavelet coefficients truncated to k or $k+1$ bit planes, respectively. Also, let $R(W_{i,\sigma(i)}^k)$ and $R(W_{i,\sigma(i)}^{k+1})$ be the bit rates required to code $W_{i,\sigma(i)}^k$ and $W_{i,\sigma(i)}^{k+1}$ losslessly. Note that $R(W_{i,\sigma(i)}^k) \leq R(W_{i,\sigma(i)}) < R(W_{i,\sigma(i)}^{k+1})$.

The basic idea of our algorithm is to only use complete bit planes for prediction. Thus, if we reach our target MSE mid-bit plane, we now have to decide whether to trim our prediction back to the last full bit plane. Alternatively, if we are close enough to the end of the current bit plane, we can decide to transmit the remaining portion of the bit plane in order to have a better predictor.

In our algorithm, if Equation (4) is satisfied, k complete bit planes are selected for prediction, and the bit rate at which we transmit $W_{i,\sigma(i)}$, $R(W_{i,\sigma(i)})$, does not change. Otherwise, $k+1$ complete bit planes are used for *both prediction and coding*. The bit rate at which we transmit $W_{i,\sigma(i)}$ must be increased to $R(W_{i,\sigma(i)}^{k+1})$. In both cases, the transmitter and re-

ceiver use the same number of complete bit planes (either k or $k + 1$) for prediction. In Equation (4), T is a threshold with typical values on the order of $0.1 - 1.0$. Note that to reduce the computational complexity, we do not look ahead to see how the prediction results propagate into the future.

$$R(W_{i,\sigma(i)}) - R(W_{i,\sigma(i)}^k) \leq T(R(W_{i,\sigma(i)}^{k+1}) - R(W_{i,\sigma(i)}^k)) \quad (4)$$

For example, the bit rate required to code the difference band 35 of the Cuprite image to the target MSE of 100 is 1.5 bpp. This corresponds to stopping mid-bit-plane in bit plane number 13. The bit rates required to code this difference band to 12 and 13 bit planes are 0.64 bpp and 1.89 bpp, respectively. If our threshold in Equation (4) is $T = 0.2$, we use 13 bit planes for prediction and encode the difference band to 13 bit planes for transmission (1.89 bpp).

However, in the case of the difference band 69, the bit rate required to code it to the target MSE of 100 is 0.16 bpp. This corresponds to stopping mid-bit-plane in bit plane number 6. The bit rates required to code this difference band to 5 and 6 bit planes are 0.04 bpp and 0.73 bpp, respectively. For the same threshold $T = 0.2$ in Equation 4, we use 5 bit planes for prediction and encode the difference band to 0.16 bpp for transmission.

Figure 9 further describes the prediction and encoding processes. If k bit planes are used for prediction, the transmitter sends $W_{i,\sigma(i)}$ at bit rate $R(W_{i,\sigma(i)})$. The receiver decodes to $\hat{W}_{i,\sigma(i)}$, takes the inverse wavelet transform to obtain $\hat{D}_{i,\sigma(i)}$ and adds to $P_{i,\sigma(i)}$, the prediction of the current band, to compute the decompressed band \hat{B}_i . However, to form the prediction of the current band for possible later use, both the transmitter and receiver truncate $W_{i,\sigma(i)}$ and $\hat{W}_{i,\sigma(i)}$ to $W_{i,\sigma(i)}^k$, take the inverse wavelet transform to obtain $D_{i,\sigma(i)}^k$, and then add $D_{i,\sigma(i)}^k$ to $P_{i,\sigma(i)}$ to compute the decompressed truncated band B_i^k which is stored.

If $k + 1$ bit planes are used for prediction, the encoder transmits $W_{i,\sigma(i)}^{k+1}$ at bit rate $R(W_{i,\sigma(i)}^{k+1})$. The receiver decodes to $\hat{W}_{i,\sigma(i)} = W_{i,\sigma(i)}^{k+1}$, takes the inverse wavelet transform to obtain $\hat{D}_{i,\sigma(i)} = D_{i,\sigma(i)}^{k+1}$ and adds to $P_{i,\sigma(i)}$, the prediction of the current band, to compute the decompressed band \hat{B}_i . What differs from the previous case of using k bit planes for prediction is that to form the prediction of B_i for possible later use, here, both the encoder and receiver simply inverse transform $W_{i,\sigma(i)}^{k+1}$ to obtain $D_{i,\sigma(i)}^{k+1}$ which is added to $P_{i,\sigma(i)}$ to compute B_i^{k+1} .

5 Results

In this section we present the results. We start by estimating reduction in the complexity of the transmitter. Then we compare the bit plane-synchronized closed loop prediction with the standard closed loop. Finally, we comment on the universality of the prediction coefficients and band ordering.

5.1 Reduction in Transmitter Complexity

In order to estimate the impact of the techniques presented in the paper, we can consider an existing FPGA-based implementation of the SPIHT engine [40], which does not include any prediction. This system achieves a 100Mbytes/second throughput rate on a system of 3 Virtex 2000E FPGAs. This performance was achieved primarily through a set of small alterations to the SPIHT algorithm that introduce minor ($< 0.2\text{db}$) degradations in image quality yet allow for aggressive pipelining of the bit plane encoder. The first FPGA performs 4 separate wavelet transforms in parallel to boost the throughput rate, and is 62% full. The other two FPGAs handle the bit plane encoding, and are 34% and 98% full respectively.

The hardware impact of including the proposed bit plane-synchronized prediction technique is quite reasonable. The inverse wavelet transform is very similar to the forward wavelet transform, and would likely also achieve a 100 Mbytes/second throughput and consume about 62% of an FPGA. The linear predictor (a multiplier, an adder, and a subtractor), as well as the truncator (replacing some bits of the wavelet coefficients with zeros) are very simple to implement in hardware, and would easily fit into the remaining space on that FPGA. In contrast, the bit plane decoder engine required by the standard closed loop would likely consume 2 additional FPGAs, and it would be quite difficult to implement at the desired throughput. The high performance of the bit plane encoder was achieved by carefully altering the algorithm to remove many sequential dependencies, and allow individual coefficients to be encoded almost independently from the others. It is likely that the same transformations could not be accomplished in the bit plane decoder. While the encoder transforms a well-structured dataset (wavelet coefficients in an array in memory) to a convoluted representation (the SPIHT encoded data), the decoder must start with the convoluted representation, and may not be able to find much parallelism at all. Our estimate is that this change would result in a 10-fold decrease in throughput if the decoder were required, compared to the approaches given here. This factor of 10 is based on the fact that the parallel processing of the bit plane encoding boosted the encoder performance by approximately a factor of 10.

Note that in a software implementation of SPIHT we can imagine an easy alternative to the bit plane-synchronized approach. As we compress the image, we maintain an array, one location per wavelet coefficient, containing information about how much of that coefficient has been transmitted. Each time the SPIHT algorithm emits a bit, the corresponding coefficient locations are updated to the current bit plane. Then, once we have emitted the desired number of bits to meet a maximum file size, we stop. The coefficient array then holds the information about the data that have been transmitted, and we can quickly create a prediction frame from the original frame and the coefficient array we have maintained. However, because of the parallel implementation of SPIHT on the FPGAs, the information needed to determine which bit plane is last for each coefficient is difficult to maintain. Therefore, the proposed bit plane synchronized algorithm is much more feasible than this approach.

5.2 Compression Results of the Bit Plane-Synchronized Closed Loop vs. Standard Closed Loop Prediction

In Figure 10, we compare the standard closed and new bit plane-synchronized predictive coders. Over a range of target MSEs from 50 to 500, the bit rate of the bit plane-synchronized coder is only slightly higher than the bit rate of the standard closed loop technique. For a target MSE of 150, the average bit rate for the proposed method is 0.52 bpp, which is an 37% increase in bit rate over the 0.38 bit rate for the closed loop prediction. However, for a target MSEs of 200 and 250, the bit rates are very close. Hence, the bit plane-synchronized loop is a very promising method to code hyperspectral data. It achieves a very good compression ratio with a low MSE and has a much lower computational complexity compared to the original closed loop prediction.

In Table 2, we test our algorithm on four different image sets to see if our results are consistent across different image sets. As the table shows, across the four image sets, the bit rate for the synchronized closed-loop is never more than 22.7% higher than standard closed loop for a target MSE of 100.

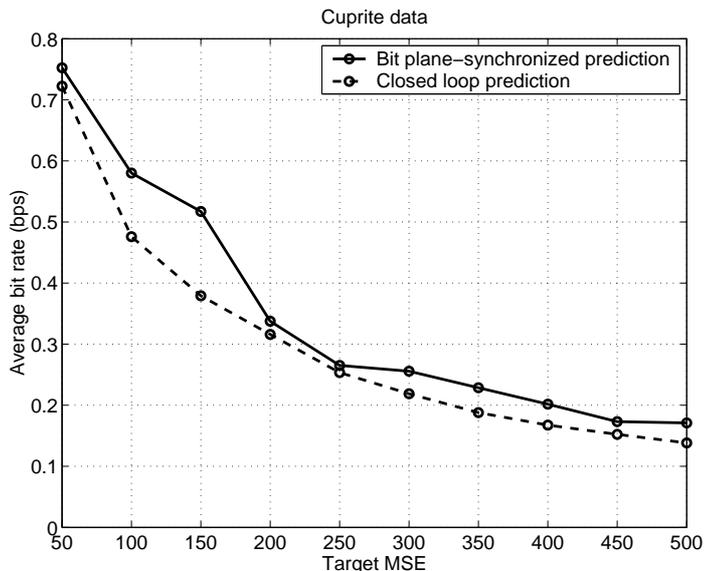


Figure 10: Bit rate vs. target MSE for bit plane-synchronized closed loop prediction and standard closed loop prediction.

5.3 Universality

Next, we investigated the universality of the prediction ordering and of the prediction coefficients (the a_{ij} s and c_{ij} s). We designed four best reverse predictors using images from the Cuprite, Jasper Ridge, Lunar Lake, and Moffett Field sites. For each of these four images, we compared the compression performance when using prediction ordering and coefficients obtained from this image (custom predictor) against using predictors designed for the other

Table 2: Average bit rate (bpp) when all bands are encoded to a target MSE of 100.

Image Data	No Prediction	Closed Loop	Bit Plane-Synchronized Closed Loop
Cuprite	1.997	0.476	0.580
Jasper Ridge	2.728	0.654	0.778
Lunar Lake	1.889	0.524	0.643
Moffett Field	2.794	0.695	0.823

images.

Table 3: Average bit rate for all bands encoded to a target MSE of 100 when either a custom predictor or predictors designed on other image sets are used.

	Training Data			
	Cuprite	Jasper Ridge	Lunar Lake	Moffett Field
Cuprite	0.477	0.561	0.513	0.534
Jasper Ridge	0.765	0.655	0.706	0.700
Lunar Lake	0.547	0.566	0.524	0.560
Moffett Field	0.714	0.721	0.710	0.697

Table 3 shows the average bit rate required to compress the four data sets to a target MSE of 100 when using different predictors. As expected, the best performance is obtained when an image is compressed using a custom predictor. However, when a predictor designed on a different image set is used, the performance is often not affected that severely. The increase in bit rate when a non-custom predictor is used is on the order of 3.4 - 17.6%.

We also compared the prediction ordering. When the Cuprite image was compressed using predictors designed using the Jasper Ridge, Lunar Lake and Moffett Field data, only 9-16 out of the 224 bands were predicted from different previous bands compared to the custom predictor. Thus we believe that good hyperspectral image compression can be obtained with a fixed band ordering and set of prediction coefficients.

6 Conclusions

In this research, we have investigated different methods of using prediction to code hyperspectral data. As expected, combining prediction with a state-of-the-art image compression algorithm significantly improves the compression ratio (from 8:1 to 33:1 for the Cuprite image and a target MSE of 100). We have also proposed a new method of prediction, the bit plane-synchronized closed loop prediction. We showed that under the constraints of a simple implementation on-board the satellite, it offers excellent performance. For example, in the case of the Cuprite data set, the presented method requires only one additional FPGA with no change in throughput compared to the standard SPIHT implementation, while the

standard closed loop would require three additional FPGAs and cause a 10-fold decrease in throughput. The price paid in compression performance is on the order of about 20% increase in the bit rate required to encode all of the bands to a target MSE of 100, compared to standard closed loop prediction.

References

- [1] “What is EOSDIS?.” Web page at <http://spsosun.gsfc.nasa.gov/eosinfo/EOSDI-Site/index.html>.
- [2] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.
- [3] T. W. Fry and S. Hauck, “Hyperspectral image compression on reconfigurable platforms,” in *IEEE Symposium on Field-Programmable Custom Computing Machines*, pp. 251–260, 2002.
- [4] A. Miguel, A. Askew, A. Chang, S. Hauck, R. Ladner, and E. Riskin, “Reduced complexity wavelet-based predictive coding of hyperspectral images for FPGA implementation,” in *Proceedings Data Compression Conference*, pp. 469–478, 2004.
- [5] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, N. J.: Prentice-Hall, 1984.
- [6] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*. New York: Chapman & Hall, 1996.
- [7] V. Cuperman and A. Gersho, “Adaptive differential vector coding of speech,” in *Conference Record GlobeCom 82*, pp. 1092–1096, Dec. 1982.
- [8] P.-C. Chang and R. M. Gray, “Gradient algorithms for designing predictive vector quantizers,” *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 34, pp. 679–690, Aug. 1986.
- [9] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer Academic Publishers, 1992.
- [10] H.-M. Hang and J. W. Woods, “Predictive vector quantization of images,” *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 33, pp. 1208–1219, Nov. 1985.
- [11] R. M. Gray, “Vector quantization,” *IEEE ASSP Magazine*, vol. 1, pp. 4–29, Apr. 1984.
- [12] G. Motta, F. Rizzo, and J. A. Storer, “Compression of hyperspectral imagery,” in *Proceedings Data Compression Conference*, pp. 333–342, Mar. 2003.
- [13] S.-E. Qian, A.-B. Hollinger, D. Williams, and D. Manak, “Vector quantization using spectral index-based multiple subcodebooks for hyperspectral data compression,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 3, pp. 1183–1190, 2000.
- [14] M. J. Ryan and M. R. Pickering, “An improved M-NVQ algorithm for the compression of hyperspectral data,” in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2, pp. 600–602, 2000.
- [15] G. P. Abousleman, T.-T. Lam, and L. J. Karam, “Robust hyperspectral image coding with channel-optimized trellis-coded quantization,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 4, pp. 820–830, 2002.

- [16] H. S. Lee, N.-H. Younan, and R. L. King, “Hyperspectral image cube compression combining JPEG 2000 and spectral decorrelation,” in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 6, pp. 3317–3319, 2000.
- [17] X. Tang, S. Cho, and W. A. Pearlman, “Comparison of 3D set partitioning methods in hyperspectral image compression featuring an improved 3D-SPIHT,” in *Proceedings of the Data Compression Conference*, p. 449, 2003.
- [18] 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 *Proceedings Data Compression Conference*, pp. 479–488, 2004.
- [19] T. Markas and J. Reif, “Multispectral image compression algorithms,” in *Proceedings of the Data Compression Conference*, vol. 3, pp. 391–400, 1993.
- [20] G. P. Abousleman, M. W. Marcellin, and B. R. Hunt, “Hyperspectral image compression using entropy-constrained predictive trellis coded quantization,” *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 566–573, 1997.
- [21] P.-L. Dragotti, G. Poggi, and R. P. Ragozini, “Compression of multispectral images by three-dimensional SPIHT algorithm,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 1, pp. 416–428, 2000.
- [22] N. D. Memon, “A bounded distortion compression scheme for hyper-spectral data,” in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2, pp. 1039–1041, 1996.
- [23] A. Rao and S. Bhargava, “Multispectral data compression using bidirectional interband prediction,” *IEEE Trans. on Geoscience and Remote Sensing*, vol. 34, no. 2, pp. 385–397, 1996.
- [24] S. R. Tate, “Band ordering in lossless compression of multispectral images,” *IEEE Transactions on Computers*, vol. 46, pp. 477–483, Apr. 1997.
- [25] M. J. Ryan and J. F. Arnold, “A suitable distortion measure for the lossy compression of hyperspectral data,” in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 4, pp. 2056–2058, 1998.
- [26] S. Tang, J. Wang, and K. Zhang, “Spatial and spectral decorrelation of lossless data compression for LANDSAT-TM imagery,” in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 1, pp. 332–334, 1994.
- [27] J. M. Spring and G. G. L. Jr., “Lossless compression of multispectral images with interband prediction error deltas,” in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 1, pp. 586–590, 1994.
- [28] R. E. Roger and M. C. Cavenor, “Lossless compression of AVIRIS images,” *IEEE Transactions on Image Processing*, vol. 5, no. 5, pp. 713–719, 1996.
- [29] M. J. Ryan and J. F. Arnold, “The lossless compression of AVIRIS images by vector quantization,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, pp. 546–550, May 1997.
- [30] S.-E. Qian, A. B. Hollinger, and Y. Hamiaux, “Study of real-time lossless data compression for hyperspectral imagery,” in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 4, pp. 2038–2042, 1999.
- [31] B. Aiazzi, P. Alba, L. Alparone, and S. Baronti, “Lossless compression of multi/hyper-spectral imagery based on a 3-D fuzzy prediction,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 5, pp. 2287–2294, 1999.

- [32] M. R. Pickering and M. J. Ryan, “Efficient spatial-spectral compression of hyperspectral data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 7, pp. 1536–1539, 2001.
- [33] S. Hauck, “The roles of FPGAs in reprogrammable systems,” *Proceedings of the IEEE*, vol. 86, pp. 615–638, Apr. 1998.
- [34] V. D. Vaughn and T. S. Wilkinson, “System considerations for multispectral image compression designs,” *IEEE Signal Processing Magazine*, vol. 12, pp. 19–31, January 1995.
- [35] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001. Second edition.
- [36] J. Edmonds, “Optimum branchings,” *Journal of Research of the National Bureau of Standards*, vol. 71B, pp. 233–240, 1967.
- [37] R. E. Tarjan, “Finding optimum branchings,” *Networks*, vol. 7, pp. 2–35, 1977.
- [38] H. N. Gabow, Z. Galil, T. Spencer, and R. E. Tarjan, “Efficient algorithms for finding minimum spanning trees in undirected and directed graphs,” *Combinatorica*, vol. 6, no. 2, pp. 109–122, 1986.
- [39] P. Kopylov and P. Franti, “Optimal layer ordering in the compression of map images,” in *Proceedings of the Data Compression Conference*, pp. 323–332, 2003.
- [40] T. Fry, “Hyperspectral image compression on reconfigurable platforms,” Master’s thesis, University of Washington, Seattle, WA, June 2001.