

Predictive Coding of Hyperspectral Images ^{*}

Agnieszka C. Miguel^{*} Richard E. Ladner[‡]
Eve A. Riskin[†] Scott Hauck[†] Dane K. Barney[‡]
Amanda R. Askew[‡]
Alexander Chang[‡]

^{*} Department of Electrical and Computer Engineering, Box 222000,
Seattle University, Seattle, WA 98122-1090

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

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

Abstract

Algorithms for lossless and lossy compression of hyperspectral images are presented. To greatly reduce the bit rate required to code images and to exploit the large amount of inter-band correlation, linear prediction between the bands is used. Each band, except the first one, is predicted by previously transmitted band. Once the prediction is formed, it is subtracted from the original

^{*}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 Richard Ladner, University of Washington, Box 352500, Seattle, WA 98195-2500, (206) 543-9347, ladner@cs.washington.edu.

band, and the residual (difference image) is compressed. To find the best prediction algorithm, the impact of various band orderings and measures of prediction quality on the compression ratios is studied. The resulting lossless compression algorithm displays performance that is comparable with other recently published results. To reduce the complexity of the lossy predictive encoder, a bit plane-synchronized closed loop predictor that does not require full decompression of a previous band at the encoder is proposed. The new technique achieves similar compression ratios to that of standard closed loop predictive coding and has a simpler implementation.

1 Introduction

Every day, NASA and other agencies collect large amounts of hyperspectral data. For example, one Airborne Visible InfraRed Imaging Spectrometer (AVIRIS) alone can produce data that require up to 16 Gbytes of storage per day. The hyperspectral images are used to identify, measure, and monitor constituents of the Earth's surface and atmosphere [1].

This huge amount of data presents a compression challenge. In this research, we propose algorithms to code the hyperspectral data. 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 a standard compression algorithm.

To optimize the prediction algorithm we study several methods of ordering the bands for prediction. To rearrange the bands into a particular ordering, we define a measure of prediction quality, the prediction mean squared error. We compute the optimal ordering using this measure as well as two restricted orderings in which each band can be predicted by the best predictor among all of the previous or future bands in the standard band numbering. In addition, we define two simple orderings in which each band is predicted by its immediate previous or future neighbor. We use the prediction mean squared error to compare those orderings.

The first proposed algorithm is lossless, that is the decompressed images are exact replicas of the original data. The difference images are encoded using bzip2 data compression algorithm [2]. We use bzip2 because it is a state-of-the-art open-source lossless data coding algorithm. We compare our results for five standard hyperspectral images with recently published results and conclude that our algorithm achieves comparable compression ratios.

The second algorithm is lossy and therefore, the decompressed image is an approximation of the original image. In this case we encode the difference image using the Set Partitioning in Hierarchical Trees (SPIHT) algorithm [3], which is a wavelet-based lossy compression technique that codes images with both high compression ratio 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) [4] and therefore has great potential for applications where the compression is performed in hardware on the aircraft and satellite platforms. Note that 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 at the transmitter, which increases its complexity. In this article we present a new prediction technique, *bit plane-synchronized closed loop prediction*, that significantly reduces the complexity of the encoder [5]. 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 transmitter to be less complex because, while it must still do an inverse wavelet transform, full decompression 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 Sections 3 and 4, we describe our prediction methodology. The algorithm for lossless predictive coding of hyperspectral images is presented in Section 5. In Section 6, we introduce our new reduced complexity lossy encoder. Finally, we conclude in Section 7.

2 Background

In this section, we present related work. We first review hyperspectral images, the bzip2 and SPIHT algorithms, and predictive coding. Then, we discuss prior work in lossless and lossy hyperspectral image compression.

2.1 Hyperspectral Images

Hyperspectral images are obtained using imaging spectrometers. Such hyperspectral sensors obtain a continuous spectrum of electromagnetic radiation reflected from the surface of the Earth. 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 one wavelength from the particular location on the Earth. The upper layers in the image cube (higher number bands) correspond to the data collected in the longest wavelengths and the bottom layers (lower number bands) correspond to the shortest wavelengths. A hyperspectral image contains several hundred narrow and contiguous wavelength bands.

Applications of hyperspectral imaging include mineral exploration, environmental monitoring, and military surveillance. Image processing tools are used to extract detailed information from hyperspectral images. Most often the objective of those algorithms involves target detection, material mapping and identification, and mapping details of surface properties [6]. After adjustments for sensor, atmospheric, and terrain effects are applied, the reflected spectral data are compared, and matched, to spectral data of known absorption features stored in libraries. Because hyperspectral data are spectrally overdetermined, it is possible to iden-

tify and distinguish between spectrally similar materials, for example, between different types and even conditions of vegetation.

Most imaging spectrometers are used aboard air-based platforms. The NASA's Airborne Visible InfraRed Imaging Spectrometer (AVIRIS) is flown on the ER-2 jet at 20km above the ground or on the Twin Otter turboprop at 4km above the ground. It produces 224 bands with wavelengths from 400 to 2500 nanometers (nm) at a resolution of 20×20 meters (ER-2) and 4×4 meters (Twin Otter). The images are 614 pixels wide and about 2000 pixels high. The output of the spectrometer consists of 12-bit floating point values which are later on scaled and rounded into signed 16-bit integers [1]. The 16-bit images are stored in scenes of 614×512 each. Typically, there are five scenes for each image. In this paper we test our algorithms on the four AVIRIS images available for download at NASA's JPL [1]. The five scenes in each image were concatenated to form the test data.

The Hyperspectral Digital Imagery Collection Experiment (HYDICE) operated by the Naval Research Lab is an imaging spectrometer that performs measurements in the same 400-2500 nm range but in 210 bands. It is capable of achieving resolution of 0.8 to 4 meters if flown at an altitude in the range from 5,000 to 25,000 feet [7]. The Probe-1 sensor is operated by Earth Search Sciences, Inc. and measures 128 wavelengths in the same 400-2500 nm range at a spatial resolution of 5-10 meters [8]. ITRES Research developed two hyperspectral sensors. The CASI-2 is capable of measuring up to 228 bands in the range of 400 to 1000 nm at a spatial resolution of 0.5-10 m. The CASI-3 sensor measures up to 288 bands in the range of 400 to 1050nm at a spatial resolution of 0.5-10 m [9]. The HyMap sensor from Integrated Spectronics measures 100-200 wavelengths in the visible to infrared spectral range [10]. The Group for Environmental Research Earth Resources Exploration Consortium operates a variety of hyperspectral sensors capable of imaging up to 76 bands in the range from 400 to 12500 nm [11]. Finally, the AISA sensors from Spectral Imaging measure up to 244 bands in the range 400-2400 nm [12].

There are only a few spaceborne hyperspectral sensors. The Hyperion sensor on-board the EO-1 satellite launched by the NASA Goddard

Space Center is capable of resolving 220 spectral bands (from 0.4 to 2.5 m) with a 30-meter resolution [13]. The FTHSI sensor on the MightySat II satellite from Air Force Research Lab measures 256 wavelengths in the range of 350 to 1050 nm.

There is a trade-off between spatial and spectral resolution in all of the hyperspectral sensors: as the number of bands increases there is a corresponding decrease in spatial resolution. Therefore, most pixels are mixed pixels, i.e. they contain spectral contribution from neighboring pixels, and the size of most targets of interest is sub-pixel.

2.2 Lossless Image Compression

Lossless compression guarantees the recovery of an exact replica of the original data and can only provide limited compression ratios, usually on the order of 2:1 to 3:1 for natural images.

2.2.1 Bzip2

Bzip2 is a recently developed state-of-the-art lossless data compression algorithm [2]. It encodes files using a block-sorting text compression algorithm based on the Burrows-Wheeler transform [14] followed by Huffman coding.

2.3 Lossy Image Compression

In lossy compression algorithms, the original image cannot be perfectly recovered. Instead, the decompressed image is an approximation of the original image. Lossy compression algorithms provide a mechanism for a controlled loss of information to ensure that the quality of the reconstructed image is adequate for the particular application. Typical compression ratios for gray scale natural images with no noticeable difference to the eye between the original and recovered images are on the order of 10:1 to 20:1.

2.3.1 Set Partitioning in Hierarchical Trees

SPIHT is a progressive lossy 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 [3]. 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.

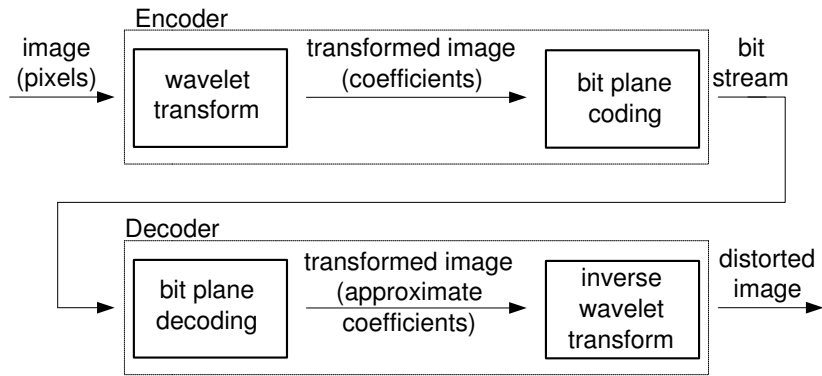


Figure 1: Block diagram of SPIHT.

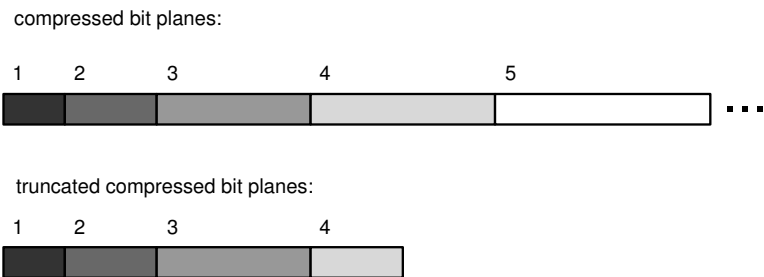


Figure 2: Bit plane coding.

2.4 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) [15] uses prediction to improve performance over standard PCM. The MPEG video coding standard [16] uses temporal prediction to significantly improve compression ratios over independent coding of video frames. Predictive vector quantization (VQ) [17, 18, 19, 20] exploits spatial correlation over a larger region of an input image or speech signal to give improvements over memoryless VQ [21, 19]. 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) [18].

2.5 Previous Work in Hyperspectral Image Compression

The proposed techniques for lossy coding of hyperspectral images can be classified into two types: vector quantization [22, 23] and transform-based [24, 25, 26, 27, 28, 29] algorithms. Qian et al. [22] generated separate subcodebooks for regions with similar spectral characteristics. Ryan and Pickering [23] used mean-normalized VQ followed by the discrete-cosine transform (DCT) in the spatial and spectral domains and entropy coding.

Markas and Reif [27] 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. Lee, Younan, and King [25] used different 1-D transforms to obtain spectral decorrelation (KL, DCT, and the difference pulse-coded modulation (DPCM)) and applied JPEG2000 to the resulting data. Abousleman, Marcellin, and Hunt in [28] proposed using DPCM for spectral decorrelation and 2D DCT for spatial decorrelation combined with entropy-constrained trellis

coded quantization (ECTCQ).

Tang, Cho, and Pearlman compared the performance of several 3D versions of SPIHT on hyperspectral data with the performance of the JPEG2000 algorithm [26]. The methods 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 [29] 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 [30] 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 [31] used simple block-based linear inter-band prediction followed by a block-based DCT. To take advantage of linear prediction between bands, Tate in [32] 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 [33]. 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.

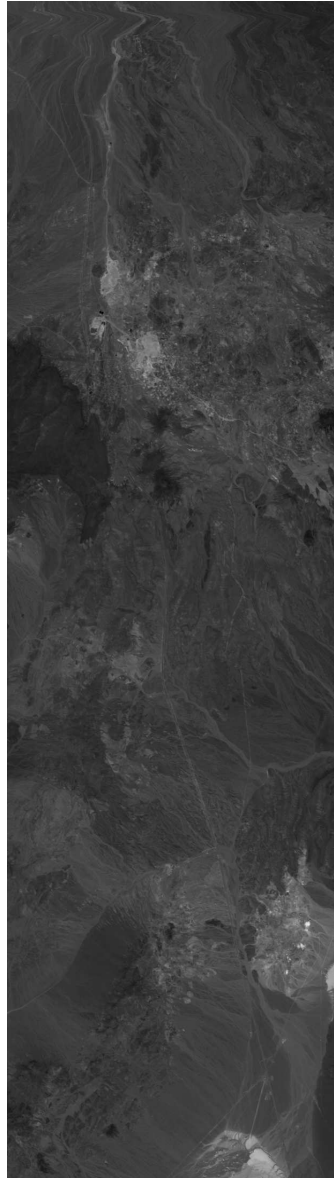
In addition to lossy compression, lossless hyperspectral image coding has been also widely investigated. Motta, Rizzo, and Storer [34] designed

a product VQ with an algorithm to determine how to form subvectors across bands. In [35], Rizzo, Carpentieri, Motta, and Storer modified this method to include a low-complexity encoder. Pickering and Ryan proposed a method based on joint optimization of a mean-normalized vector quantization (proposed by Ryan and Arnold in [36]) applied in the spatial domain and a DCT in the spectral domain [37]. Aiazzi, Alba, Alparone, and Baronti compressed multi- and hyperspectral data using a 3D fuzzy prediction [38]. Qian, Hollinger, and Hamiaux applied the Consultative Committee for Space Data System (CCSDS) compression algorithm as an entropy coder to the results of several different prediction methods [39]. Adaptive DPCM with linear prediction was investigated by Roger and Cavenor [40].

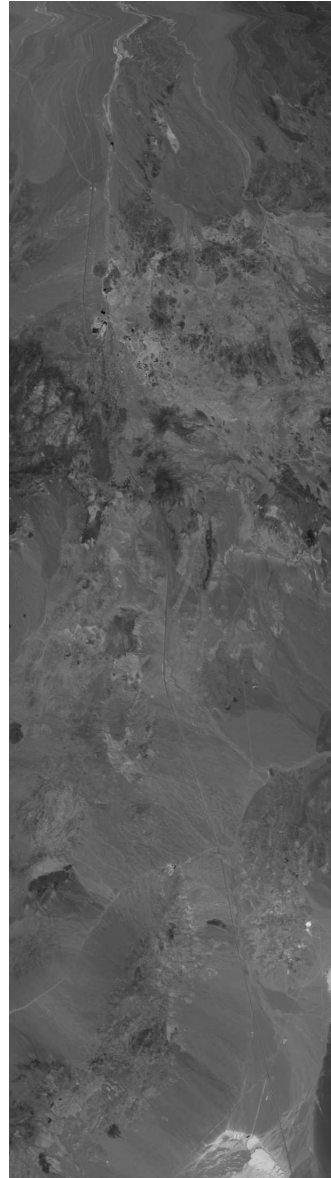
3 Prediction Methodology

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 [32, 41] and simple linear prediction has been successfully used to remove the spectral correlation between bands [28].

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 ($0 \leq j \leq m$), including 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

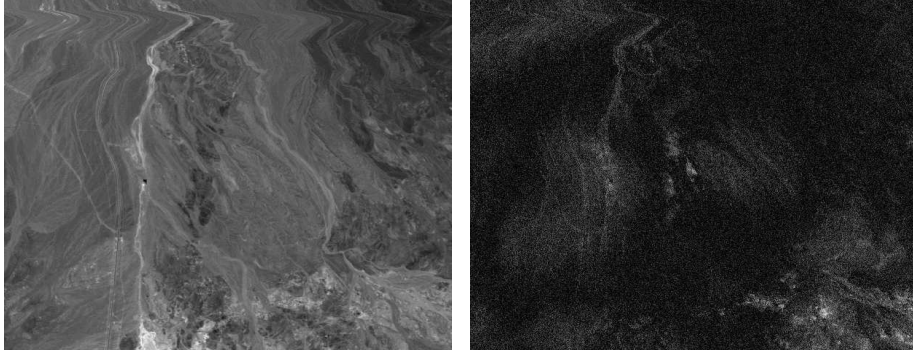


(a) Band 30



(b) Band 200

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



(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). Only the first scene is shown.

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 for the first scene of the Cuprite image. 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_{im} . 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 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 [42]. 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 defined by a function

$$\sigma : \{0, \dots, m-1\} \rightarrow \{0, \dots, m\}. \quad (2)$$

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*:

$$\frac{1}{mn} \sum_{i=0}^{m-1} \|D_{i,\sigma(i)}\|^2. \quad (3)$$

4 Band Prediction Ordering Solutions

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 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.

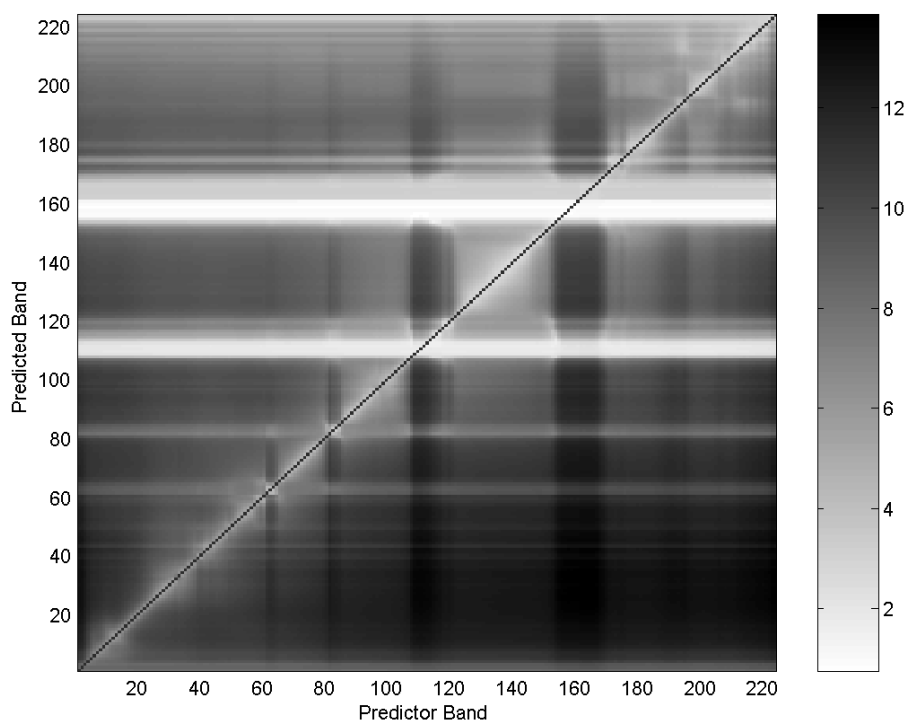


Figure 5: Prediction matrix for the Cuprite image (log scale).

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 larger 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 constraint 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 follows. 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 (3) is minimized.

The algorithm for finding the minimum spanning tree in a directed graph was first developed in the 1960s [43]. It was then applied in the 1970s [44] to solve network flow problems and its implementation was further improved in [45]. The best complexity bound for finding the minimum spanning tree in a directed graph is $O(n^2)$ [44, 45], but these algorithms are significantly more complex than computing the other orderings. The minimum spanning tree technique was also used by Tate [32] to find an optimal band ordering for lossless compression of multispectral images and by Kopylov and Fränti [46] 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 five 224-band images: Cuprite (C), Jasper Ridge (JR), Low Altitude (LA), Lunar Lake (LL), and Moffett Field (MF). These images were downloaded from the NASA’s JPL AVIRIS site at [1]. The test images are all stored as 16-bit signed integers. As can be seen, the best reverse ordering is actually very close to the opti-

mal ordering (within 19%).

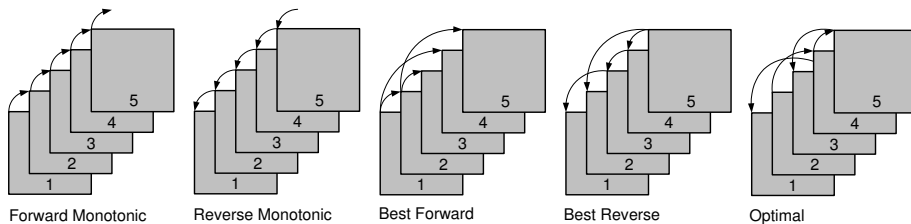


Figure 6: Examples of prediction band orderings.

Table 1: Average PMSE for five different band orderings.

Ordering	C	JR	LA	LL	MF
Forward Monotonic	367.6	1100.5	3121.1	550.2	2529.6
Best Forward	359.4	1096.9	3118.0	530.5	2443.3
Reverse Monotonic	296.4	1007.9	2609.8	424.5	2401.2
Best Reverse	264.7	1001.4	2606.4	335.0	2311.3
Optimal	263.8	875.6	2187.7	332.2	2005.9

4.1 Compression Quality Measures

In this Section we discuss the various metrics that we used to quantify the quality of the proposed lossless and lossy compression algorithms.

Average Compression Ratio The quality of a particular lossless compression scheme is measured by its average compression ratio computed by dividing the sum of all of the original file sizes by the sum of all of the compressed file sizes.

Bit Rate A common measure of compression is the bit rate, which is defined to be the file size in bits divided by the total number of pixels. For example, for the uncompressed Cuprite image, which is 16 bits per pixel (bpp), a compressed file bit rate of 2 bpp is equivalent to an 8:1 compression ratio.

Target MSE For each band i , the quality of a particular lossy 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.

Spectral MSE To evaluate the effects of our lossy compression algorithm on the shape of the spectral profile of each image pixel, we compute the spectral MSE defined as

$$\text{SMSE}(x, y) = \frac{1}{m} \sum_{i=0}^{m-1} \left(B_i(x, y) - \hat{B}_i(x, y) \right)^2 \quad (4)$$

where m is the number of bands. Its purpose is to show the extent to which the target MSE is satisfied in the spectral direction. Note that the proposed algorithm was designed to deliver the target MSE for each band but not for each spectral profile. However, we feel that it is important to test the performance of our method in the spectral domain.

Maximum Scaled Difference Error We introduce the Maximum Scaled Difference Error (MaxSDE) to facilitate direct comparison between spectral signatures of pixels in the original and decompressed images. Such comparison is important because classification of hyperspectral images is based on matching the spectral profile of each pixel with predefined spectral signatures stored in a library. If the compression algorithms modify the shape of spectral curves, some pixels may not be classified correctly. The MaxSDE of a pixel (x, y) is computed as a ratio of the maximum absolute difference between the original and decompressed pixel values and the average absolute spectrum of that pixel. The MaxSDE is defined as

$$\text{MaxSDE}(x, y) = \max_i \left\{ \frac{m |B_i(x, y) - \hat{B}_i(x, y)|}{\sum_{i=0}^{m-1} |B_i(x, y)|} \right\}. \quad (5)$$

Mean Scaled Difference Error To further evaluate the quality of our lossy compression algorithm, we also study the Mean Scaled Difference Error (MSDE). MSDE of a pixel (x, y) is defined as a ratio of the average absolute difference between the original and decompressed pixel values and the average spectrum of that pixel:

$$\text{MSDE}(x, y) = \frac{\sum_{i=0}^{m-1} |B_i(x, y) - \hat{B}_i(x, y)|}{\sum_{i=0}^{m-1} |B_i(x, y)|}. \quad (6)$$

Note that because some hyperspectral bands have very different range of pixel values, in both cases (MaxSDE and MSDE), we scale the absolute difference errors by the average value of each spectrum.

5 Lossless Coding Using Prediction

In this section we examine the benefit of using prediction for lossless compression of hyperspectral images. Previous work of Tate [32] has shown that prediction works very well for lossless compression of multispectral images. The same holds true for hyperspectral images.

5.1 Algorithms

As noted in the previous section there are several band orderings that can be used based on the PMSE metric. We saw that the best reverse ordering achieved almost the same PMSE as the optimal ordering. With lossless compression one can choose an ordering that is based on compressed file size (CFS), rather than PMSE. Unlike in the lossy setting, in the lossless setting the decoder has available for prediction any *original* band that it has already received. To compute the best forward, best reverse, and optimal orderings we first compute an $m \times (m + 1)$ matrix F where F_{ij} is the compressed file size in bytes of D_{ij} (cf. equations (1)). The difference images D_{ij} are computed using a_{ij} and c_{ij} that minimize $\|D_{i,\sigma(i)}\|^2 = \|B_{ij} - (a_{ij}B_j + c_{ij}B_m)\|^2$. We then use the same algorithms as described in Section 4 substituting the file size F_{ij} for PMSE $\|D_{i,\sigma(i)}\|^2/n$. Naturally, the forward and reverse monotonic algorithms do not change because they do not depend on PMSE.

5.2 Results

To compare the PMSE and CFS metrics and the different ordering algorithms we used the five hyperspectral images and compressed them using the different methods. All difference images were compressed using bzip2. We tested several data compression algorithms (bzip2 version 1.0.2 [2], lzp version 1 [47], shcodec version 1.0.1 [48], gzip version 1.1.3 as included with Linux ppmz [49], bwtzip version 1.0.2 [50], szip version 1.12a [51], and zzip version 0.36b [52]) and found that bzip2 performed as well as any of them. In addition, bzip2 is readily available and free on many platforms. We measure the performance of a method using compression ratio. As can be seen in Table 2, results for the compression based on CFS are consistently slightly better than those based on PMSE. As expected the optimal ordering gives the best compression ratios for the CFS metric. Somewhat surprising is that in the PMSE metric, all orderings we considered performed about the same. In fact, the best forward algorithm slightly outperforms the optimal algorithm. This can happen because in the optimal PMSE algorithm, optimality is based on finding the minimum PMSE over all orderings, not on finding the minimum file size over all orderings. From Table 1 the best reverse ordering has PMSE almost the same as the optimal ordering, while the best forward has significantly larger PMSE. However, as seen in Table 2, this difference does not translate to better lossless compression for the best reverse algorithm in either the PMSE or CFS metrics.

Table 2: Compression ratios based on PMSE and CFS metrics.

	Fwd	Rev	BestRev		BestFwd		Optimal	
			PMSE	CFS	PMSE	CFS	PMSE	CFS
Cuprite	3.09	3.08	3.10	3.22	3.09	3.21	3.10	3.23
Jasper Ridge	3.07	3.06	3.08	3.17	3.07	3.16	3.07	3.18
Low Altitude	2.88	2.87	2.90	3.02	2.89	3.01	2.88	3.03
Lunar Lake	3.08	3.06	3.09	3.20	3.07	3.20	3.08	3.21
Moffett Field	3.02	3.01	3.03	3.12	3.01	3.11	3.01	3.13

Table 3 compares using bzip2 with no prediction, with the best reverse, best forward, and optimal orderings using the CFS metric. The last column contains the results for SLSQ-OPT, which is the best algo-

rithm of Rizzo *et al* [35]. Clearly, using prediction is significantly better than using no prediction. The best forward, best reverse, and optimal algorithms perform about the same, but slightly worse than SLSQ-OPT.

Table 3: Compression ratios for various band orderings using bzip2 and CFS metric.

	No Pred	BestRev	BestFwd	Optimal	SLSQ-OPT
Cuprite	2.30	3.21	3.22	3.23	3.24
Jasper Ridge	2.12	3.16	3.17	3.18	3.23
Low Altitude	2.15	3.01	3.02	3.03	3.04
Lunar Lake	2.35	3.20	3.20	3.21	3.23
Moffett Field	2.13	3.11	3.12	3.13	3.21
Average	2.21	3.14	3.15	3.16	3.19

5.3 Universality

We investigated the problem of how sensitive the compression ratio is to the particular prediction coefficients and band ordering. We hypothesize that the prediction coefficients and band orderings are similar for a given instrument and only vary a moderate amount from image to image. If this were true then one could define a single set of prediction coefficients and a band ordering that would work well for a wide variety of images. That is, we could define *universal* prediction coefficients and band ordering. Table 4 gives the results of how sensitive the compression ratio is when varying the prediction coefficients and band ordering. In this study, for each image I , we computed the optimal prediction coefficients a_{ij}^I and c_{ij}^I minimizing $\|D_{ij}^I\|^2/n$. We then computed the optimal ordering using the CFS metric. For every two images I and J , we used the prediction coefficients and optimal ordering for J to compress image I using bzip2 on the difference images. Thus, the image J is the training image for compressing image I . As can be seen from the table, using a training image other than the image to be compressed always yields a worse compression ratio, but not remarkably worse. The worst training image is Low Altitude. In the worst case (compressing Jasper Ridge image using Low Altitude as a training image), the compression ratio

reduces from 3.027 to 2.802, which is still quite close.

Table 4: Compression sensitivity to prediction coefficients and orderings. Compression ratios are shown.

	Training Image				
	C	JR	LA	LL	MF
Cuprite	3.231	-0.107	-0.158	-0.059	-0.068
Jasper Ridge	-0.158	3.179	-0.225	-0.094	-0.080
Low Altitude	-0.152	-0.178	3.027	-0.178	-0.138
Lunar Lake	-0.067	-0.055	-0.187	3.212	-0.086
Moffett Field	-0.057	-0.060	-0.151	-0.082	3.127

6 Wavelet-Based Lossy Coding Using Prediction

In this section we examine the benefit of using prediction for lossy compression of hyperspectral images. We also investigate a low complexity implementation of the wavelet-based lossy coding algorithm. In all of the following experiments we use best forward ordering of the bands because it achieved the best compression ratios compared to the other orderings.

Prediction significantly improves the results of lossy compression. For example, as shown in Figure 7, for the Cuprite image, 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.

6.1 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 (7), the transmitter uses a decompressed previous band $\hat{B}_{\sigma(i)}$ to form $P_{i,\sigma(i)}$, the

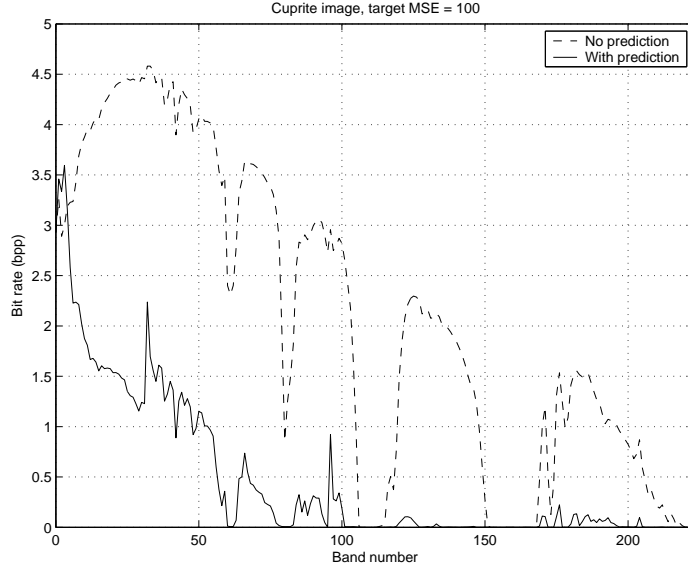


Figure 7: Comparison of bit rates in bits per pixel (bpp) required to code the Cuprite image to MSE=100 with and without prediction for best forward ordering.

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.

$$\begin{aligned}
 P_{i,\sigma(i)} &= a_{i,\sigma(i)}\hat{B}_{\sigma(i)} + c_{i,\sigma(i)}B_m \\
 D_{i,\sigma(i)} &= B_i - P_{i,\sigma(i)} \\
 \hat{B}_i &= P_{i,\sigma(i)} + \hat{D}_{i,\sigma(i)}
 \end{aligned} \tag{7}$$

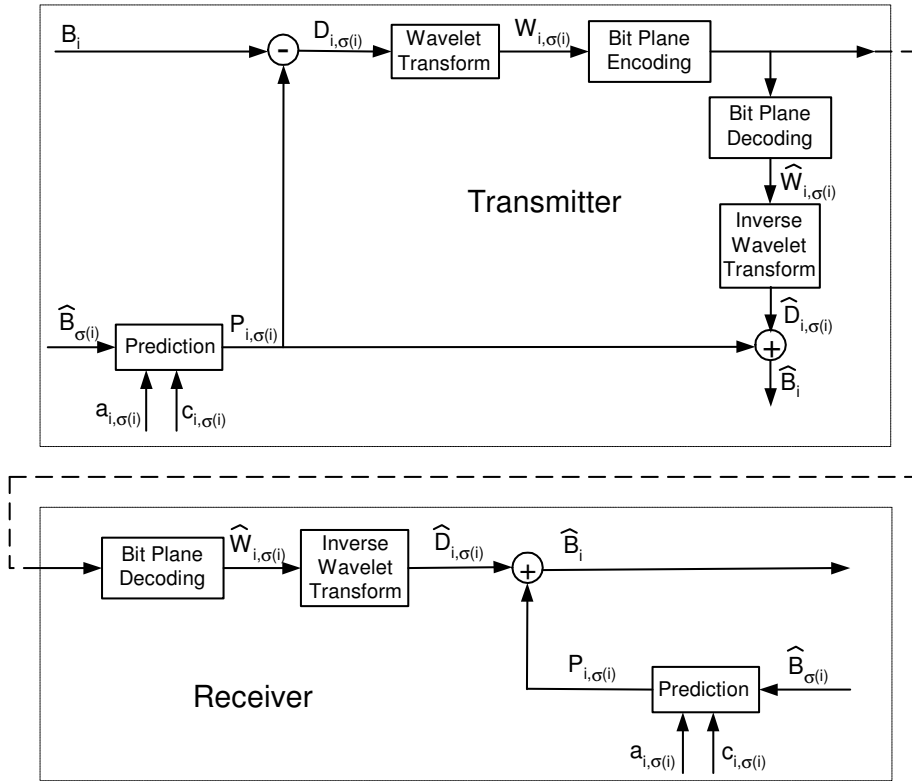


Figure 8: Standard closed loop prediction.

6.2 Bit Plane-Synchronized Closed Loop Prediction

As a lower complexity solution, 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.

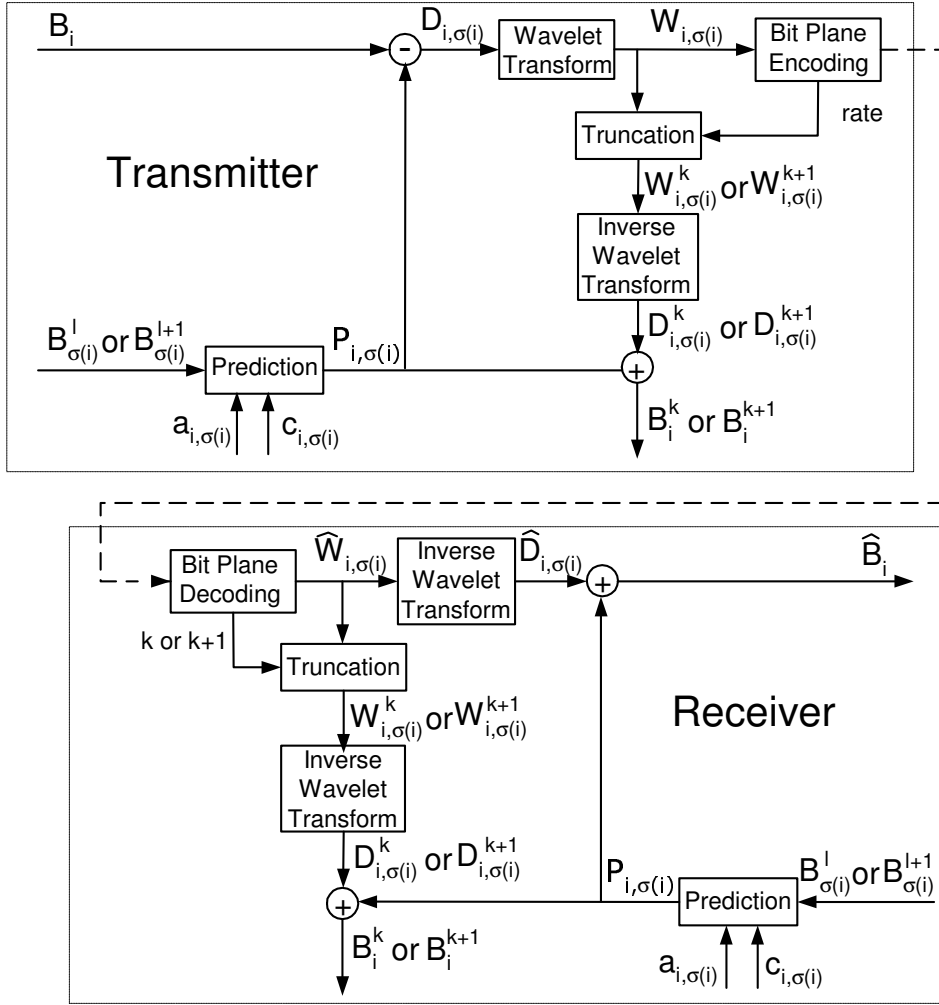


Figure 9: Bit plane-synchronized closed loop prediction.

6.2.1 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 $k+1$. 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

$$R(W_{i,\sigma(i)}^k) \leq R(W_{i,\sigma(i)}) < R(W_{i,\sigma(i)}^{k+1}). \quad (8)$$

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 (9) 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 receiver use the same number of complete bit planes (either k or $k+1$) for prediction. In Equation (9), 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.

$$R(W_{i,\sigma(i)}) - R(W_{i,\sigma(i)}^k) \leq T(R(W_{i,\sigma(i)}^{k+1}) - R(W_{i,\sigma(i)}^k)) \quad (9)$$

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 (9) 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 9, 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} .

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 encodes 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, similarly to the standard closed loop, this method would have higher computational complexity compared to the bit-plane synchronized closed loop solution. For every wavelet coefficient in every bit plane, we would have to execute one additional operation of counter increment. Therefore, the proposed bit plane synchronized algorithm is more feasible than this approach.

6.2.2 Results

In this section we present results. We first study the effects of compression on the spectral profiles of image pixels. 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.

Effects of Lossy Compression on the Shape of Spectral Profiles

For some applications, it may be important to preserve the shape of the spectral profiles of image pixels. For example, during classification, the decompressed spectral profiles may be compared to a standard spectrum shape stored in a spectral library. Our lossy compression algorithm has not been designed with this goal in mind, instead we concentrated on optimizing the rate-distortion performance using the MSE of each band as the distortion measure. However, to study the effects of lossy compression on the spectral profiles, we computed the Spectral MSE for each pixel in the five test images after they were encoded to a target MSE of 100 using the closed loop algorithm with best forward band ordering. In Table 5 we show the percentage of pixels with a spectral MSE below 50, 100, 150, 200, and 500. Note that not all pixels have their spectral MSE lower than the target MSE of 100. However, a large number of pixels (89-95%) is within 1.5 times the target MSE.

Next we calculate the Maximum Scaled Difference Error (MaxSDE) to find out how similar the shape of the decompressed spectrum is to the shape of the original spectrum. In Table 6 we show the percentage of pixels with a spectrum such that for each band the absolute difference

Table 5: Percentage of pixels with a spectral MSE below given value when all bands are encoded to a target MSE of 100.

	Spectral MSE lower than				
	50	100	150	200	500
Cuprite	7.15%	69.45%	91.62%	97.16%	99.93%
Low Altitude	2.86%	70.97%	95.26%	99.04%	99.95%
Jasper Ridge	4.49%	68.54%	93.76%	98.54%	99.94%
Lunar Lake	5.89%	66.98%	89.17%	95.67%	99.88%
Moffett Field	9.09%	65.00%	91.65%	97.81%	99.93%

between the original and decompressed pixel values is below 0.01, 0.015, 0.02, and 0.05. Note that about 90-100% of all pixels have their MaxSDE lower than 0.05. This leads us to believe that our algorithm preserves the spectral profile shape for most image pixels.

Table 6: Percentage of pixels such that the Maximum Scaled Difference Error is below a given value.

	MaxSDE lower than			
	0.01	0.015	0.02	0.05
Cuprite	1.39%	39.59%	83.59%	99.98%
Low Altitude	0.07%	3.50%	20.53%	96.35%
Jasper Ridge	0.00%	0.70%	13.36%	97.18%
Lunar Lake	7.11%	42.94%	73.14%	99.85%
Moffett Field	0.05%	2.98%	22.61%	89.71%

Finally, we compute the Mean Scaled Difference Error (MSDE) and calculate the number of pixels that have their MSDE below 0.0025, 0.005, 0.0075, 0.01, and 0.02. As shown in Table 7, at least 87% of pixels have an average absolute difference error less than 0.01. In summary, our lossy compression algorithm, although not designed to preserve the shape of the spectral profiles exhibits satisfactory performance in that area.

Table 7: Percentage of pixels such that their MSDE is below a certain value.

	MaxSDE lower than				
	0.0025	0.005	0.0075	0.01	0.02
Cuprite	0.83%	85.16%	99.19%	99.94%	100.00%
Low Altitude	0.03%	19.93%	73.84%	92.77%	98.65%
Jasper Ridge	0.00%	9.66%	69.20%	92.25%	97.56%
Lunar Lake	5.45%	71.58%	94.48%	98.93%	99.99%
Moffett Field	0.05%	20.45%	71.21%	87.11%	99.60%

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 100, the average bit rate for the proposed method is 0.51 bpp, which is an 13.3% increase in bit rate over the 0.45 bit rate for the closed loop prediction. However, for a target MSE of 200, 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 8, we test our algorithm on five different image sets to verify that our results are consistent across different images. As the table shows, across the five image sets, the compression ratio for the synchronized closed-loop is never less than 13.8% lower than standard closed loop for a target MSE of 100.

Universality Next, we investigated the universality of the prediction ordering and of the prediction coefficients (the a_{ijs} and c_{ijs}). We designed five best forward predictors using images from the Cuprite, Low Altitude, Jasper Ridge, Lunar Lake, and Moffett Field sites. For each

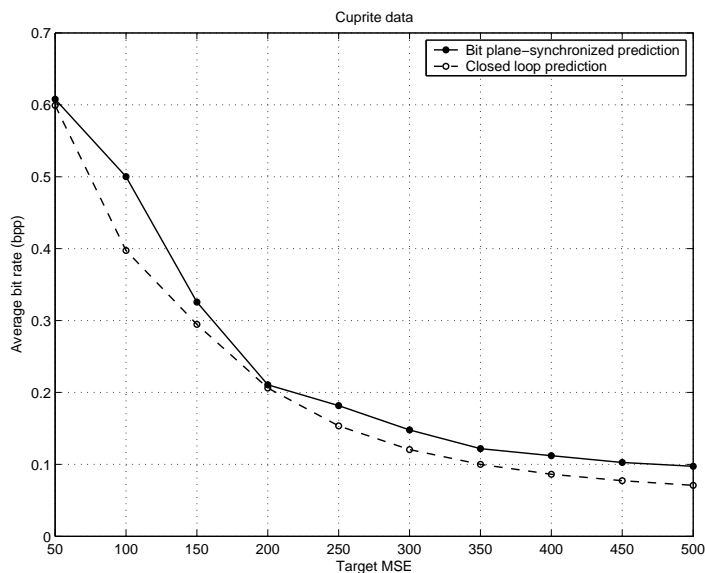


Figure 10: Average bit rate vs. target MSE for bit plane-synchronized closed loop prediction and standard closed loop prediction.

of these five 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 images.

Table 9 shows the compression ratio when the four data sets are encoded to a target MSE of 100 using different predictors. Best forward ordering and the closed loop algorithm were used for this experiment. As expected, the best performance is obtained when an image is compressed using a custom predictor. When a predictor designed on a different image set is used, the performance may be affected quite severely. The decrease in compression ratio when a non-custom predictor is used is on the order of 0 - 25%. Nonetheless, using any image as the predictor yields large compression ratios for the target MSE of 100.

We also compared the prediction ordering. When the Cuprite image was compressed using predictors designed using the Jasper Ridge, Low Altitude, Lunar Lake and Moffett Field data, only 31-42 out of the

Table 8: Compression ratio when all bands are encoded to a target MSE of 100. Best forward ordering was used for the closed and the bit plane-synchronized loop simulations.

Image Data	No Prediction	Closed Loop	Bit Plane-Synchronized Closed Loop
Cuprite	7.63	35.32	31.05
Jasper Ridge	5.77	25.58	24.13
Low Altitude	6.55	24.61	22.78
Lunar Lake	8.35	35.79	31.79
Moffett Field	6.16	24.93	23.92

Table 9: Compression sensitivity to prediction coefficients and orderings when all bands are encoded to a target MSE of 100 with a best forward ordering when either a custom predictor or predictors designed on other image sets are used. Compression ratios are shown.

	Training Image				
	C	JR	LA	LL	MF
Cuprite	35.32	-5.60	-6.37	-2.12	-4.73
Jasper Ridge	-4.09	25.58	-6.02	-2.81	-3.17
Low Altitude	-4.19	-5.62	24.61	-5.35	-4.60
Lunar Lake	-2.72	-3.67	-7.79	35.79	-5.35
Moffett Field	-0.03	-1.43	-2.52	-1.48	24.93

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.

7 Conclusion

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 lossy compression with a target MSE of 100 and from 2.3:1 to 3.2:1 for the same image and lossless compression).

We studied the impact of different band orderings on the compression ratios. We found that the lossless compression performance for both best forward and best reverse ordering is very close to that for optimal ordering. We also considered two measures of prediction quality: compressed file size (CFS) and the prediction MSE (PMSE), and analyzed their impact on the compression performance. We discovered that lossless compression ratios based on the CFS metric were consistently better than those based on the PMSE, however, the difference was very small. We concluded that the resulting lossless compression algorithm displays performance that is comparable with other recently published results.

To reduce the complexity of the lossy predictive encoder, we proposed 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 implementation.

References

- [1] J. P. Laboratory, “AVIRIS (airborne visible/infrared imaging spectrometer) homepage.” <http://aviris.jpl.nasa.gov/>.
- [2] J. Seward, “The bzip2 and libbzip2 official home page.” <http://sources.redhat.com/bzip2/>.
- [3] 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.
- [4] 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.
- [5] 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.

- [6] P. Shippert, "Why use hyperspectral imagery?," *Photogrammetric Engineering & Remote Sensing, Journal Of The American Society For Photogrammetry And Remote Sensing*, vol. 70, pp. 377–380, April 2004.
- [7] R. W. Basedow, M. Kappus, L. J. Rickard, and M. E. Anderson, "HYDICE: Operational system status." <http://1tpwww.gsfc.nasa.gov/ISSSR-95/hydiceop.htm>.
- [8] Earth Search Sciences, Inc., "Probe-1." <http://www.earthsearch.com/index.htm>.
- [9] ITRES Research, Inc., "CASI-2 and CASI-3." <http://www.earthsearch.com/index.htm>.
- [10] Integrated Spectronics, "HyMap." <http://www.intspec.com/>.
- [11] Group for Environmental Research Earth Resources Exploration Consortium, "Airborne hyperspectral and multispectral imaging systems." <http://www.ger.com/ie.html>.
- [12] Spectral Imaging, "AISA." <http://www.specim.fi/index.html>.
- [13] U.S. Geological Survey EROS Data Center, "USGS EO-1 website." <http://eo1.usgs.gov/hyperion.php>.
- [14] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm," Tech. Rep. 124, Digital Equipment Corporation, 1994.
- [15] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, N. J.: Prentice-Hall, 1984.
- [16] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*. New York: Chapman & Hall, 1996.
- [17] V. Cuperman and A. Gersho, "Adaptive differential vector coding of speech," in *Conference Record GlobeCom 82*, pp. 1092–1096, Dec. 1982.
- [18] 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.
- [19] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer Academic Publishers, 1992.
- [20] 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.
- [21] R. M. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, pp. 4–29, Apr. 1984.
- [22] 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.

- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] T. Markas and J. Reif, "Multispectral image compression algorithms," in *Proceedings of the Data Compression Conference*, vol. 3, pp. 391–400, 1993.
- [28] 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.
- [29] 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.
- [30] N. D. Memon, "A bounded distortion compression scheme for hyperspectral data," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2, pp. 1039–1041, 1996.
- [31] 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.
- [32] S. R. Tate, "Band ordering in lossless compression of multispectral images," *IEEE Transactions on Computers*, vol. 46, pp. 477–483, Apr. 1997.
- [33] 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.
- [34] G. Motta, F. Rizzo, and J. A. Storer, "Compression of hyperspectral imagery," in *Proceedings Data Compression Conference*, pp. 333–342, Mar. 2003.

- [35] 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.
- [36] 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.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] V. D. Vaughn and T. S. Wilkinson, "System considerations for multi-spectral image compression designs," *IEEE Signal Processing Magazine*, vol. 12, pp. 19–31, January 1995.
- [42] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. 2001. Second edition.
- [43] J. Edmonds, "Optimum branchings," *Journal of Research of the National Bureau of Standards*, vol. 71B, pp. 233–240, 1967.
- [44] R. E. Tarjan, "Finding optimum branchings," *Networks*, vol. 7, pp. 2–35, 1977.
- [45] 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.
- [46] P. Kopylov and P. Fränti, "Optimal layer ordering in the compression of map images," in *Proceedings of the Data Compression Conference*, pp. 323–332, 2003.
- [47] C. Bloom, "Dictionary coders (lzp)." http://www.cbloom.com/src/index_lz.html.
- [48] A. Simakov, "Shcodec home page." <http://webcenter.ru/~xander/>.

- [49] H. Peltola and J. Tarhio, "Ppmz for Linux." <http://www.cs.hut.fi/u/tarhio/ppmz/>.
- [50] S. T. Lavavej, "bwtzip." <http://nuwen.net/bwtzip.html>.
- [51] M. Schindler, "Szip homepage." <http://www.compressconsult.com/szip/>.
- [52] D. Debin, "Zzip homepage." <http://debin.org/zzip/>.