

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 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

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 almost the same compression ratio as 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 gigabytes of data per day [1]. The hyperspectral images are transmitted to the ground station, stored, and used to identify occurrences of fire, water, and snow on the Earth.

The huge amount of data generated by satellites presents a data 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

^{*}This work appeared in part in the Proceedings of the NASA Earth Science Technology Conference, 2003. 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.

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 transmitted, is predicted by another 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 from another band, 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. 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

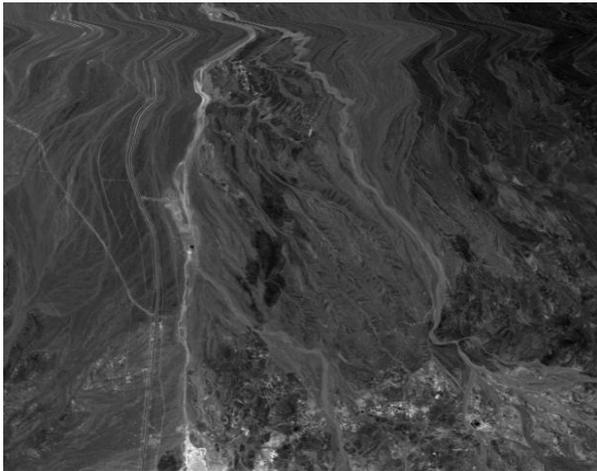
In this section, we present related work. We first review SPIHT. Then, we discuss prior work in hyperspectral image compression and finish with a discussion on FPGAs.

2.1 Set Partitioning in Hierarchical Trees

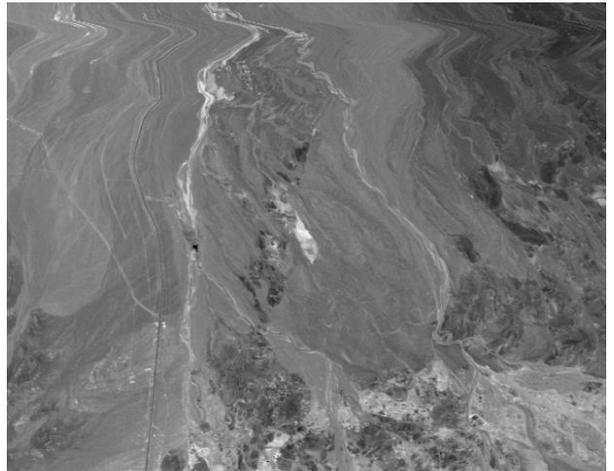
SPIHT is a progressive image coder, which first approximates an image with a few bits of data, and then improves the quality of approximation as more information is encoded. The encoder first performs a wavelet transform on the image pixels. Then, the wavelet coefficients are encoded one bit plane at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

2.2 Previous Work in Hyperspectral Image Compression

The proposed techniques for lossy compression of hyperspectral images can be classified into two types: vector quantization [4, 5, 6] and transform-based [7, 8, 9] algorithms. Motta, Rizzo, and Storer [4] designed a product VQ with an algorithm to determine how to form subvectors across bands. Linear prediction as a method to reduce inter-band correlation was investigated by Memon in [10] who proposed adaptive reordering of the spectral components



(a) Band 30



(b) Band 200

Figure 1: Sample bands 30 and 200 of a 224-band image of the Cuprite geology site.

of each pixel followed by a piecewise linear function at a specified error tolerance. Rao and Bhargava [11] used simple block-based linear inter-band prediction followed by a block-based DCT. To take advantage of linear prediction between bands, Tate in [12] explored unconstrained optimal reordering of the multispectral bands followed by linear prediction, which uses spatial neighborhoods to predict each pixel and arithmetic coding.

2.3 FPGAs

Field-Programmable Gate Arrays (FPGAs) are chips that can be programmed and reprogrammed to implement complex digital logic [13]. Current chips may run at speeds of 100MHz or more, with logic capacities in the millions of logic gates. Because of their reprogrammability, they are particularly attractive in remote applications, where configurations can be changed and upgraded by simply shipping a configuration file.

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 1 for an example). It has been suggested in the literature that the pixel values across different bands are linearly related [12, 14].

We 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 ($0 \leq j \leq m$) 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} .

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

Note that the prediction P_{im} requires that $a_{ij} = 0$ 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 prediction MSE, the more bits are needed to compress the difference. The prediction MSE 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 [15]. 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 .

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

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

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 $0 \leq i < m-1$. 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 prediction MSE.

Figure 2 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.

To take advantage of the fact that some bands are better predictors than other 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

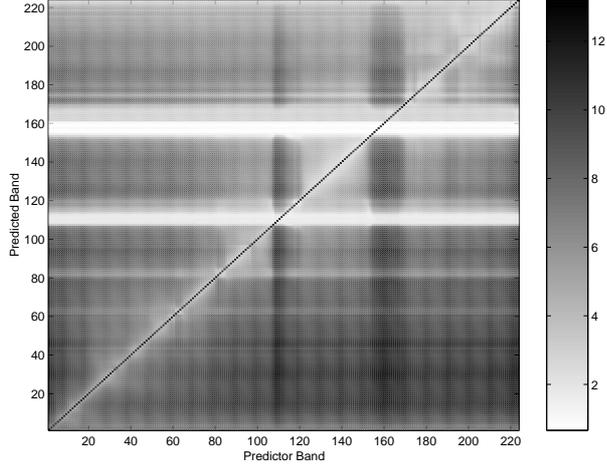


Figure 2: Prediction Matrix (Log Scale).

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 optimal ordering as a graph problem (finding the minimum weight rooted spanning tree on a directed graph [16, 12]).

The five possible orderings are shown in Fig. 3. Table 1 lists the average prediction MSE 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 0.32%). Since the running time of the best reverse ordering is much faster than the optimal, we use it for all simulations in this paper.

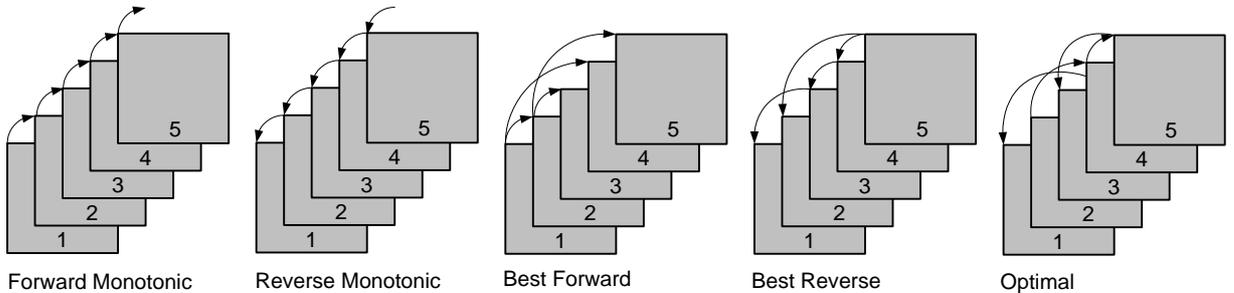


Figure 3: Prediction band ordering.

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

Ordering Type	Average prediction MSE
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, 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 40:1.

3.3 Standard Closed Loop Prediction

To predict the current band, a previous band is needed. In *closed loop prediction*, shown in Figure 4, the decompressed version of a previously encoded band is used for prediction by both the encoder and decoder.

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 significantly increases computational complexity.

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

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 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

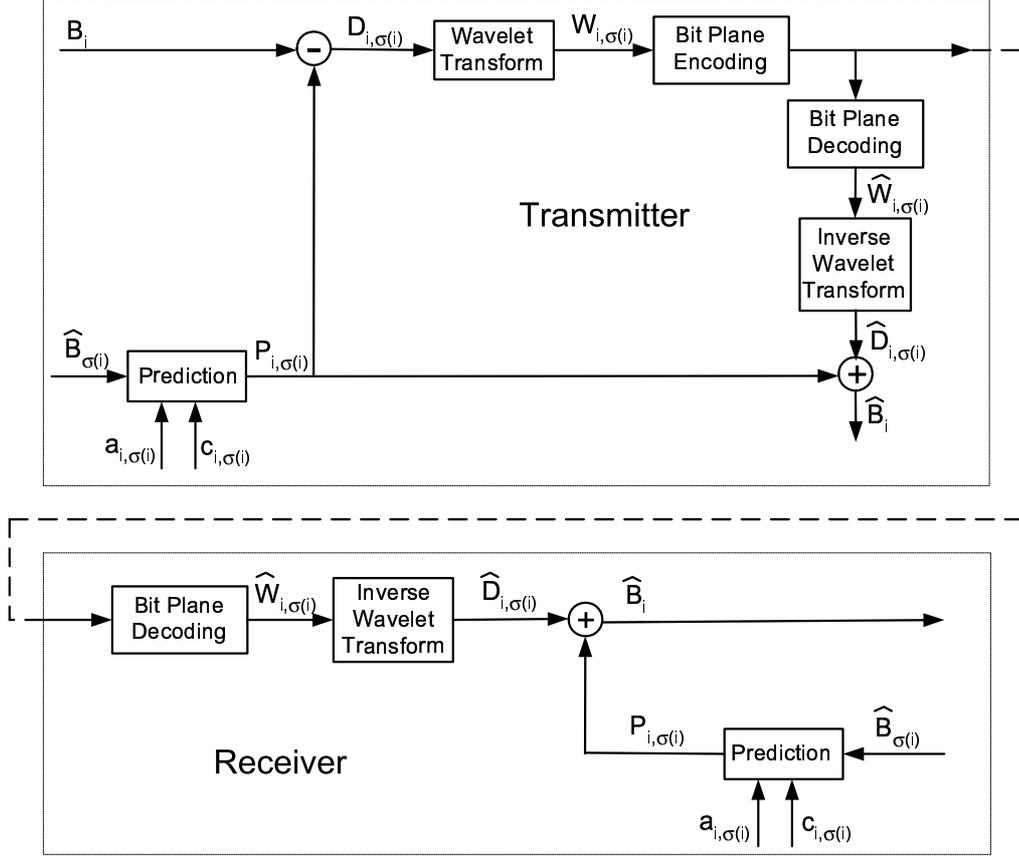


Figure 4: Standard closed loop prediction.

corresponds to stopping the encoder mid-bit-plane, for example, in bit plane number k . 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 $R(W_{i,\sigma(i)}^k) \leq R(W_{i,\sigma(i)}) < R(W_{i,\sigma(i)}^{k+1})$.

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 receiver 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 12. 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.

However, in the case of the difference band 69, the bit rate required to code it to the

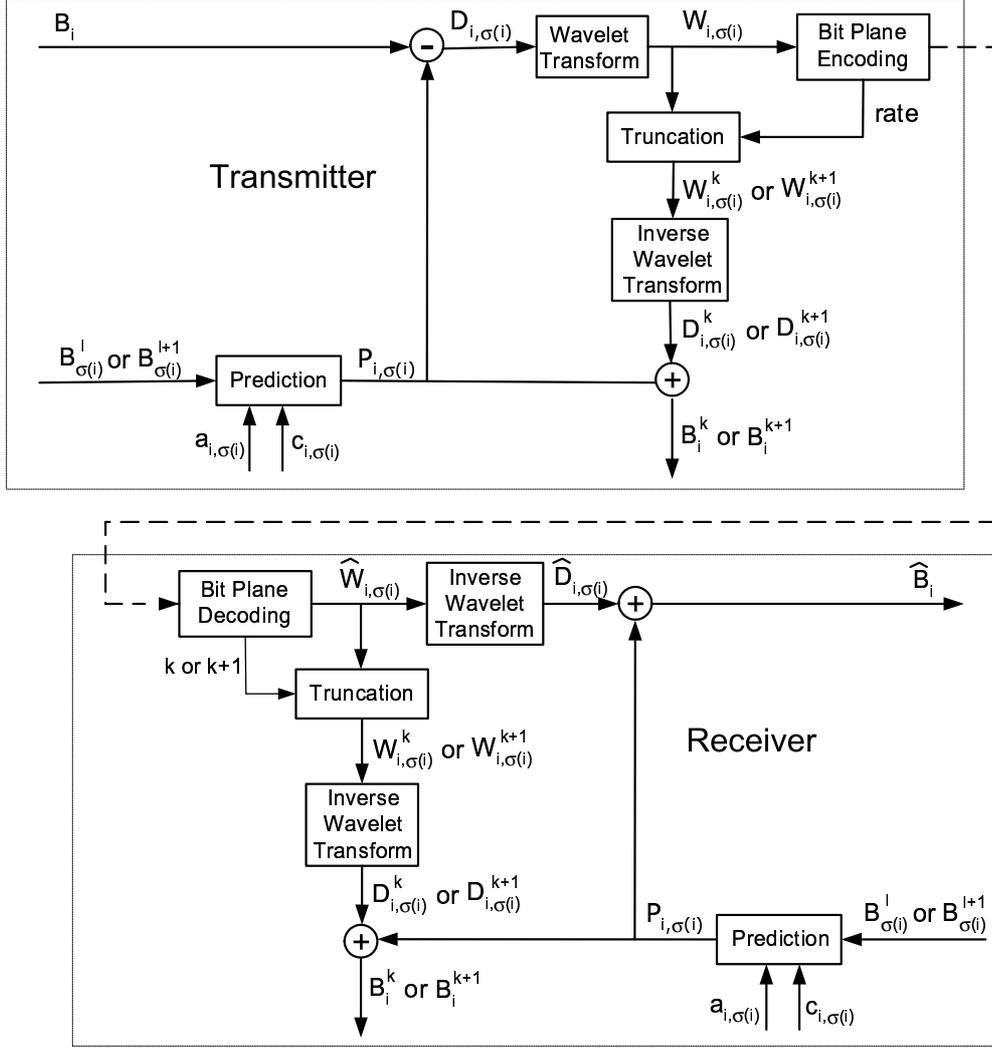


Figure 5: Bit plane-synchronized closed loop prediction.

target MSE of 100 is 0.16 bpp. This corresponds to stopping mid-bit-plane in bit plane number 5. 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 5 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 encoder and receiver truncate $W_{i,\sigma(i)}$ or $\hat{W}_{i,\sigma(i)}$ 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 Figure 6, 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 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 0.65 bpp which is an 18% increase in bit rate over the 0.55 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.

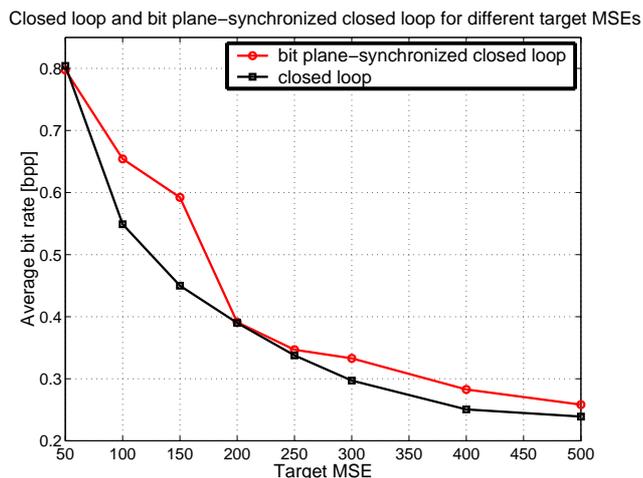


Figure 6: Bit rate vs. Target MSE for bit plane-synchronized closed loop prediction and standard closed loop prediction.

6 Conclusions

In this research, we have investigated different methods of using prediction to code hyperspectral data. As expected, we saw that combining prediction with a state-of-the-art image compression algorithm significantly improves the compression ratio. We have also proposed a new method of prediction, the bit plane-synchronized loop. We showed that under the constraints of a simple implementation on-board the satellite, it offers excellent performance.

References

- [1] “What is EOSDIS?.” Web page at <http://spsosun.gsfc.nasa.gov/eosinfo/EOSDIS-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] G. Motta, F. Rizzo, and J. A. Storer, “Compression of hyperspectral imagery,” in *Proceedings Data Compression Conference*, pp. 333–342, Mar. 2003.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] S. R. Tate, “Band ordering in lossless compression of multispectral images,” *IEEE Transactions on Computers*, vol. 46, pp. 477–483, Apr. 1997.
- [13] S. Hauck, “The roles of FPGAs in reprogrammable systems,” *Proceedings of the IEEE*, vol. 86, pp. 615–638, Apr. 1998.
- [14] 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.
- [15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. 2001. Second edition.
- [16] 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.