Efficient wavelet-based predictive Slepian–Wolf coding for hyperspectral imagery ☆

Ngai-Man Cheung[a],[*], Caimu Tang[b], Antonio Ortega[a], Cauligi S. Raghavendra[a],[b]

[a]*Department of Electrical Engineering-Systems Division, University of Southern California, Los Angeles, CA 90089-2560, USA*
[b]*Department of Computer Science, University of Southern California, Los Angeles, CA 90089-0781, USA*

Abstract

Hyperspectral imagery is usually highly correlated, in some cases within each spectral band, but in particular across neighboring frequency bands. In this paper, we propose to use distributed source coding (DSC) to exploit this correlation with an eye to a more efficient hardware implementation. The theoretical underpinnings of DSC are laid out in the pioneering work of Slepian and Wolf, and Wyner and Ziv, which provide bounds on the achievable compression when encoding correlated sources with side information available at the decoder. We apply DSC principles to hyperspectral images by encoding individual images (each image representing a spectral band) under the assumption that these bands are correlated. Using DSC tools allows us to operate in "open loop" at the encoder, so that encoding a band does not require having access to decoded versions of (spectrally) neighboring bands. We first compute the parameters of a linear predictor to estimate the current spectral band from a neighboring one, and estimate the correlation between these two bands (after prediction). Then a wavelet transform is applied and a bit-plane representation is used for the resulting wavelet coefficients. We observe that in typical hyperspectral images, bit-planes of same frequency and significance located in neighboring spectral bands are correlated. We exploit this correlation by using low-density parity-check (LDPC)-based Slepian–Wolf codes. The code rates are chosen based on the estimated correlation. We demonstrate that set partitioning of wavelet coefficients, such as that introduced in the popular SPIHT algorithm, can be combined with our proposed DSC techniques so that coefficient significance information is sent independently for all spectral bands, while sign and refinement bits can be coded using DSC. Our proposed scheme is appealing for hardware implementation as it is easy to parallelize and has modest memory requirements. In addition to these implementation advantages, our scheme can achieve competitive coding performance. Our results for high-correlation spectral bands from the NASA AVIRIS dataset show, at medium to high reconstructed qualities, gains of up to 5dB as compared to encoding the spectral bands independently using SPIHT. Our proposed techniques are also competitive compared to 3D wavelet coding methods, where filtering is applied spatially within each spectral band, as well as across spectral bands.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Hyperspectral imagery; Distributed source coding; Slepian–Wolf coding; SPIHT; Correlation estimation; LDPC

1. Introduction

Hyperspectral image data consists of hundreds of spectral bands, leading to very large raw data size. For example, the images captured by AVIRIS
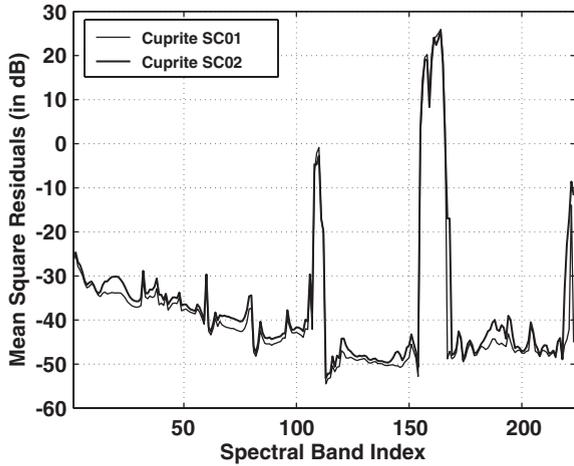
Fig. 1. Mean-square residuals after simple image alignment and subtraction.

(Airborne Visible/Infrared Imaging Spectrometer, operated by NASA) include 224 spectral bands, so that a single hyperspectral image contains up to 140 Mbytes of raw data [2]; therefore, efficient compression is necessary for practical hyperspectral imaging applications. In addition, hyperspectral images are usually captured by satellites that use embedded processors with limited resources, so encoding complexity is critical.

In a hyperspectral dataset many spectral bands are highly correlated. This is shown in Fig. 1, where image mean-square residuals after simple alignment are shown for two different views on a site. Neighboring bands tend to be correlated and the degree of correlation varies relatively slowly over a broad range of spectral regions. Thus, exploiting inter-band correlation using, for example, inter-band prediction followed by 2D compression [3] or 3D wavelet decompositions [4] has proven to be a popular approach to compress hyperspectral images.[1]

In this paper we propose novel compression techniques for hyperspectral imagery that requires low encoding complexity while achieving competitive compression performance. Our proposed techniques use wavelet-based encoding to enable lossy to lossless, scalable encoding of the spectral bands. This is combined with distributed source coding

(DSC) techniques [5], which are used to exploit the inter-band correlation. Slepian and Wolf [5] proved that two correlated sources can be optimally encoded even if the encoder only has access to the two sources separately, as long as both encoded streams are available at the decoder. This counter-intuitive result permits (in principle) significant complexity reductions at the encoder, where low complexity is most needed for hyperspectral imagery, while preserving the encoder's ability to optimally compress the data (approaching the same performance as conventional schemes based on predictive framework) by exploiting the redundancy in the correlated sources. Slepian–Wolf coding was a dormant niche of information theory for nearly three decades, until the recent development of low-complexity, capacity-approaching (turbo or low-density parity-check (LDPC)) channel codes. Now practical applications seem feasible and codes for the Slepian–Wolf problem have been proposed by a number of authors [6–9]. Applications of Slepian–Wolf coding include data aggregation in sensor networks [10,11] and video coding, e.g., [12,13]. In the video coding application, the correlated sources are successive video frames. In this paper, correlated sources will be successive bands of hyperspectral imagery.

Our proposed scheme, set partitioning in hierarchical trees with Slepian–Wolf coding (SW-SPIHT), is an extension of the well-known SPIHT algorithm [14]. SW-SPIHT first uses an iterative set-partitioning algorithm to extract bit-planes. Bit-planes at the same bit position in neighboring bands are shown to be correlated. Once the first spectral band, which is encoded independently, is available to the joint decoder, bit-planes can be extracted from it and successive bit-planes at corresponding subbands and significance levels from the second spectral band can be decoded. All bit-planes other than those from the first spectral band are encoded independently using an LDPC-based Slepian–Wolf code [9,15] and jointly decoded by a sum–product decoding algorithm. As an example of coding performance, for the NASA AVIRIS hyperspectral images data set, at medium to high quality, SW-SPIHT can achieve similar coding efficiency compared to 3D-SPIHT, and up to 5dB gain compared to 2D-SPIHT on individual bands. Note that when all bit-planes are encoded SW-SPIHT can also provide lossless compression. In many applications of hyperspectral images preserving the spectral signature is important (e.g., the spectral signature

---

[1]As will be illustrated later, it is easy to modify an algorithm that exploits crossband correlation so that it operates independently in each frame when correlation is low, as is the case in some spectral regions in Fig. 1.

may be used for classification and preserving classification rates becomes important [4]). SW-SPIHT provides flexibility in the choice of operating points, so that the rate can be selected in order to preserve the spectral signature. A detailed analysis is presented in Section 5, which demonstrates that SW-SPIHT can provide a more uniform distortion profile across bands than 3D wavelet techniques. This is shown to be advantageous in terms of preserving the spectral signature.

To the best of our knowledge we are the first to propose the application of DSC techniques in the context of hyperspectral imagery [1]. Another key novelty of our work is that we combine (i) DSC techniques operating on binary data and (ii) bit-plane successive refinement encoding based on set partitioning, a technique that has been broadly used in wavelet-based image coding. These two techniques achieve coding efficiency by exploiting different characteristics of the input data, namely, spatial and frequency localization of wavelet coefficient energy (set partitioning) and correlation across spectral bands (DSC). We show that by combining these techniques, so that DSC is applied when it provides the most gain, a better performance is achieved than if DSC were applied directly to "raw" bit-planes (i.e., complete bit-planes, rather than set-partitioned ones). More specifically, our proposed codec relies on standard set-partitioning techniques to signal the location of "significant" wavelet coefficients, while using DSC to encode signs and refinement bits.

Note that DSC techniques require the encoder to have information about the correlation between the source being encoded and side information available at the decoder. In our application, the side information, i.e., neighboring bands, is actually available at the encoder and thus correlation can be estimated exactly. However, to estimate this correlation accurately may involve a significant overhead, in terms of memory and complexity at the encoder. Thus, another important novelty in our work is that we take into account the cost involved in estimating inter-band correlation. We propose low-cost techniques for correlation estimation and demonstrate that these result in minimal losses in compression performance. Our proposed approach has potential advantages when compared with competing techniques that exploit crossband correlation, such as inter-band predictive methods and 3D wavelet techniques.

In *inter-band prediction* approaches [3], a band is predicted using previously encoded bands and the resulting prediction residuals are encoded using standard image coding techniques. Compared to inter-band prediction approaches, our proposed DSC approach has the following advantages. First, inter-band prediction methods need to generate exact copies of the decoded bands at the encoder, so encoders need to perform decoding as well, and decoding complexity could be significant, e.g., comparable to encoding complexity. In contrast, DSC requires only access to correlation statistics and these statistics can be reliably estimated with low complexity from uncoded data, as will be shown. Second, inter-band predictive methods are inherently serial, since each band is encoded based on a predictor obtained from previously decoded bands. We will show that a DSC approach has the potential to enable parallel encoding once the inter-band correlations have been estimated. While correlation estimation requires data exchange across bands, this process is much simpler than encoding/decoding. This inherent parallelism can facilitate hardware implementations and greatly increases the on-board encoding speed. Third, our proposed approach facilitates scalability. We apply DSC to bit-planes extracted from wavelet coefficient data. A given bit-plane in a given subband depends only on the same bit-plane in a neighboring spectral band. Thus, once hyperspectral data have been encoded, efficient rate scalability can be achieved by decoding all spectral bands up to the same bit-plane resolution level. In contrast, "closed-loop" inter-band prediction makes it difficult to achieve efficient rate scalability. Note that this problem is analogous to that of achieving scalability in a video compression scenario, for which DSC techniques have also been proposed recently [16,17].

3D wavelet methods, including 3D-SPECK and 3D-SPIHT [4], provide an alternative to predictive techniques. 3D wavelet methods can also exploit inter-band correlation by performing filtering across spectral bands, with the expectation that most of the signal energy will be concentrated in low-pass subbands (corresponding to low spatial and "cross-band" frequencies). A drawback of these methods is that they lead to complex memory management issues. A naive implementation would consist of loading several spectral bands in memory so as to perform crossband filtering. More sophisticated approaches are possible, e.g., loading simultaneously only subbands corresponding to a given spatial frequency in various spectral bands, but these approaches have the drawback of requiring

numerous iterations of memory read and write. In contrast, a DSC approach requires storing in memory a single spectral band at a time, once correlation statistics are estimated. These lower memory requirements could potentially lead to lower power consumption at the encoder, since a substantial amount of off-chip memory access would be avoided. This is particularly important because off-chip memory accesses often consume up to one order of magnitude higher power than on-chip data accesses [18].

This paper is organized as follows. We present the proposed codec in Section 2 and our prediction and estimation model in Section 3. Implementation and experimental results are described in Section 4 followed by a discussion in Section 5 and conclusions in Section 6.

2. Codec architecture

Consider two hyperspectral bands, X and Y, and denote $\hat{X}$ the reconstruction of band X at the decoder, which will be used to produce the side information to decode Y.[2] This side information, $\hat{X}'$, is generated by linear prediction $\hat{X}' = \alpha\hat{X} + \beta$, where α and β will be estimated at the encoder.

Let us assume first that the correlation statistics are known to both the encoder and the decoder. In particular, assume that for every set of binary data to be encoded (e.g., a bit-plane or part of a bit-plane extracted from Y), we have access to the "crossover probabilities", i.e., the probabilities that two bits in corresponding bit-plane positions of $\hat{X}'$ and Y, respectively, are not equal. These crossover probabilities will tend to be different at each level of significance (i.e., crossover probability will tend to increase from MSB to LSB bit-planes). Section 3 will present techniques to efficiently estimate both crossover probabilities and prediction parameters from input data; these techniques require processing a small fraction of pixels in spectral bands X and Y so that computation overhead is kept low.

In our work we use SPIHT [14], a well-known wavelet-based image coding algorithm, as a starting point. Similar ideas could be applied to other image coding algorithms that achieve successive refine-

---

[2]Note that, as will be discussed later, decoding is possible with many reconstructions of X at the decoder; as coarser versions of X are used, the reconstruction of Y will be correspondingly coarser. This facilitates rate scalability, i.e., multiple operating points can be achieved with a single embedded bitstream.

ment of information by representing data in bit-planes. At each pass, SPIHT uses a significance test on wavelet coefficients to partition them into two sets: the significant set and the insignificant set. Bits corresponding to significance information are entropy coded and output by the encoder; they allow the decoder to update the list of coefficients in the significant set.

A block diagram of our proposed system is shown in Fig. 2. Band X is encoded and decoded independently (i.e., without information from any other band) using a wavelet transform and SPIHT coding. The reconstructed band $\hat{X}$ will then be used to form side information to decode Y. As for band Y, the first step is again a wavelet transform $T(f, n)$ where f is the filter used in the transform and n is the number of transformation levels. Then SW-SPIHT successively updates the set of significant wavelet coefficients of Y at each pass. As shown in Fig. 3, at the end of each iteration, a sign bit-plane, a refinement bit-plane and corresponding significance bits are generated. Sign bits and refinement bits are
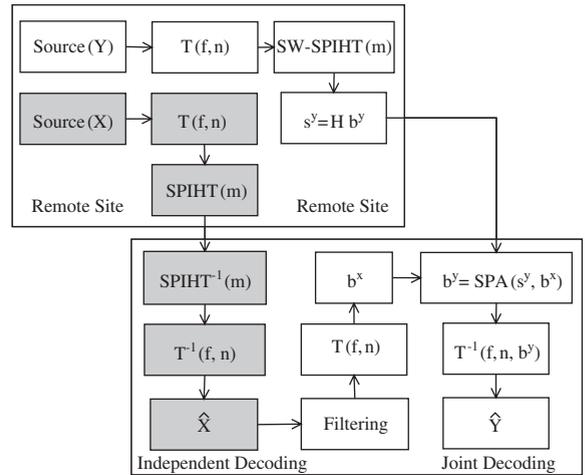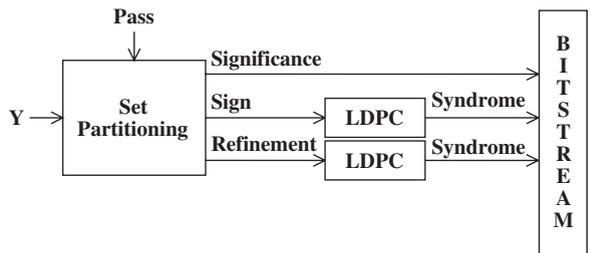

Fig. 2. Block diagram of SW-SPIHT.

Fig. 3. Bit-plane coding in SW-SPIHT.

encoded using an LDPC-based Slepian–Wolf code and corresponding syndrome bits are output to the bitstream. However, significance bits are encoded independently, i.e., exactly as they would have been coded in a standard SPIHT approach. In Section 5 we will provide some insights to explain why in our proposed system coding of bit-planes *after* set partitioning tends to outperform raw bit-plane coding, i.e., without set partitioning.

In what follows, b^w, b_i^w, and $b_i^w(l)$ denote a bit-plane, the ith bit-plane and the lth bit of the ith bit-plane of image W, respectively. Also in what follows, unless otherwise stated, bit-planes are sets of sign bits and refinement bits as generated after set partitioning at a given level of significance. This is illustrated by Fig. 3. The encoder comprises the following steps (see Fig. 4).

(E-1) Estimation of predictor coefficients α and β using a subset of information in X and Y.

(E-2) Application of the prediction coefficients to obtain wavelet transform coefficients of X'.

(E-3) Computation of wavelet transform of Y.

(E-4) At each iteration, set partitioning of the wavelet coefficients of Y to extract bit-planes b_i^y ($1 \leqslant i \leqslant m$).

(E-5) Application of the significance tree of Y to the wavelet coefficients of X' to extract bit-planes b_i^x ($1 \leqslant i \leqslant m$).

(E-6) Computation of $\hat{p}_i$, estimated crossover probability of the bit-plane pair (b_i^x, b_i^y) ($1 \leqslant i \leqslant m$) of X' and Y respectively.

(E-7) Determination of the Slepian–Wolf coding rate based on the estimated crossover probability.

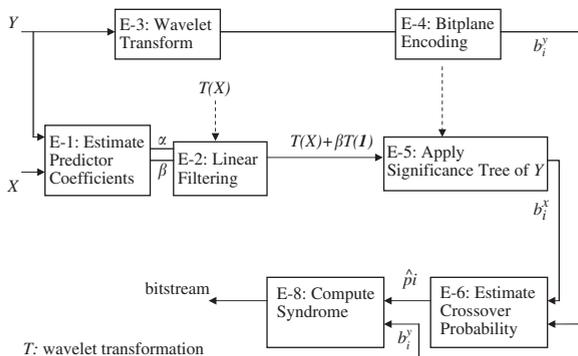(E-8) Generation of parity-check matrix for b_i^y ($1 \leqslant i \leqslant m$).

Fig. 4. Encoding using proposed system.

The compressed bitstream generated for Y includes, for each coding pass, the corresponding significance map and the syndromes generated for sign and refinement bit-planes. Note that in this algorithm it is not necessary to have access to an encoded version of X. Also, we will discuss in Section 3 how prediction coefficients and crossover probabilities can be estimated with low complexity.

At the decoder, the reconstructed $\hat{X}'$ is transformed using $T(f, n)$, i.e., the same wavelet transformation used on Y at the encoder. Then the significance tree of Y (*not* X) is used to parse the wavelet coefficients of $\hat{X}'$ in order to extract the bit-planes to be used as side information. Note that the significance tree is sent to the decoder directly (i.e., coded in "intra" mode) and thus will be available without requiring any side information. This is an important aspect of our algorithm because we have chosen to partition Y into sets before applying Slepian–Wolf coding techniques to some of the data. Thus, in order to produce the "right" side information for decoding we must apply the *same set partitioning* to $\hat{X}'$. The LDPC sum–product algorithm (SPA) is used to decode the bit-planes of Y given syndrome bits and side-information bit-planes from $\hat{X}'$.

When all bit-planes are decoded and coefficients have been refined to a desired quality level, the decoder applies the inverse wavelet transform $T^{-1}(f, n)$ to reconstruct $\hat{Y}$ an estimate of Y. Since Slepian–Wolf coding is used to code these bit-planes, they can be transmitted with no or negligible information loss, as long as the correlation model is correct. Information loss would only occur if some of the crossover probabilities were underestimated. Note also that simple quality scalability can be achieved with our scheme; since any bit-plane in Y is encoded based on a single bit-plane in X, we can scale the rate by stopping the bit-plane refinement at the same level of significance in both X and Y. SW-SPIHT can also provide lossless compression for hyperspectral imagery when all bit-planes are coded, provided that an integer-to-integer wavelet transform [19] is used. Note that the least significant bit-planes tend to be uncorrelated from image to image and also have near maximum entropy; thus, in lossless applications, these bit-planes can be sent uncoded.

Crossover probabilities are used by the encoder to determine the compression rate. This rate determines which parity-check matrix should be used for a given bit-plane. In SW-SPIHT, irregular Gallager

codes are used. A table is built offline that associates different crossover probabilities with random seeds for proper parity-check matrices. Once the cross-over probability between a bit-plane and its corresponding side-information bit-plane is obtained, a proper parity-check matrix can be selected at run-time. To make sure the same parity-check matrix is used at the decoder, the random seed used by the encoder to generate the parity-check matrix is sent to the decoder. To match the exact bit-plane width, column puncturing and splitting is used on the parity-check matrix.

In summary our decoder comprises the following steps:

(D-1) Application of prediction coefficients to obtain $\hat{X}' = \alpha \hat{X} + \beta$.

(D-2) Transformation of $\hat{X}'$ using the same wavelet transform used for Y at the encoder.

(D-3) Application of the significance tree of Y to the wavelet coefficients of $\hat{X}'$ to extract m bit-planes b_i^x ($1 \leqslant i \leqslant m$).

(D-4) Computation of a priori probability $Pr(b_i^y(j) = 0 | b_i^x(j))$ for $b_i^x(j) = 0$ or 1. Decoding of Y using SPA.

Note that our proposed technique can be also extended to support multiple sources of side information. For example, if we consider encoding each bit-plane of the current band, n, which we denote X_n, using the corresponding bit-planes in the two previous bands, $n-1$ and $n-2$, denoted X_{n-1} and X_{n-2}, respectively, we could in theory achieve an encoding rate close to $H(X_n|X_{n-1}X_{n-2})$, and this would be smaller than that of using only single side information, $H(X_n|X_{n-i})$, $i = 1, 2$. This would require a minimal increase in complexity at the encoder (due to computation of additional prediction coefficients and crossover probabilities) but would lead to an increase in decoder complexity. We tested this approach for the datasets considered in this paper, and observed that the gains may not justify the additional complexity at the decoder except lossless or near-lossless coding operation. For most bit-planes, using band $n-1$ alone as side information already leads to significant compression gains, and a relative small conditional entropy, $H(X_n|X_{n-1})$. In our observation, the additional compression gain when using X_{n-2} as additional side information, i.e., $H(X_n|X_{n-1})-$
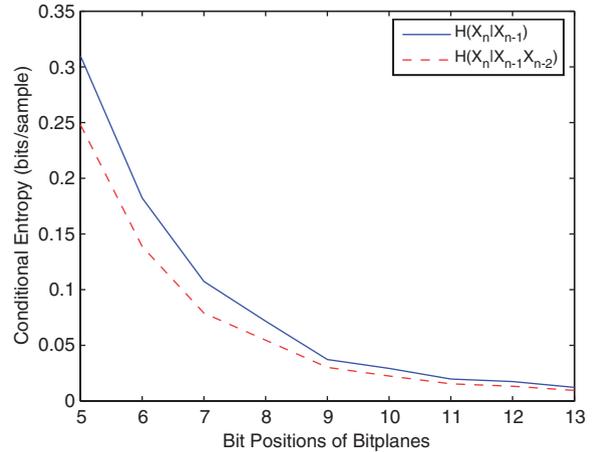

Fig. 5. Example of conditional entropy of different bit-planes.

$H(X_n|X_{n-1}X_{n-2})$, tends to be relatively small. As an example, Fig. 5 shows $H(X_n|X_{n-1})$ and $H(X_n|X_{n-1}X_{n-2})$ at different bit-planes of typical spectral bands. As shown in the figure, the reduction in coding rate achievable when using multiple bands as side information is only around 0.01 bits/sample in the more significant bit-planes, which for many lossy compression applications would not justify the additional complexity at the decoder. As for the less significant bit-planes, the reduction in conditional entropy when using multiple bands as side information is larger (up to 0.05 bits/sample), so that in lossless or near-lossless scenarios multiple side information may be useful. Given that we are not focusing specifically in the near-lossless or lossless case, the rest of this paper describes our design and experimental results based on a single band used as side information.

3. Correlation estimation and encoder complexity comparison

The performance of DSC techniques depends strongly on the estimation of correlation and prediction parameters. In our system, we need to estimate two sets of parameters, namely, (i) the linear prediction coefficients, α and β, and (ii) the bit-plane crossover probabilities. In this section, we demonstrate that accurate estimation of correlation parameters can be achieved using techniques involving a limited number of data transfers and computations. Because this estimation is accurate and requires low complexity, our proposed DSC techniques compare favorably with inter-band predictive approaches, which usually involve a

substantial amount of data transfer (e.g., if a whole spectral band is predicted using another spectral band, then all pixels in the predictor image need to be fetched in order to generate a prediction residue). Reduction in the amount of data transfer is particularly important for applications operating in embedded environments, such as hyperspectral imagery compression in satellites. In these applications the encoder may only have enough internal memory to accommodate the current spectral band (since the application programs and operating systems may have occupied significant portions of the internal memory). In order to perform prediction, the system would need to fetch the relevant information from neighboring bands, which is likely to be stored in external memory. Such external memory accesses usually lead to substantial power consumption and delay. For example, while some sophisticated CPU/DSPs can handle multiple arithmetic operations in a single cycle, accessing external memory data may incur latency of the order of tens of cycles [20]. So it is desirable to reduce the total amount of data exchanged, which translates into reduction in overall system complexity.

In what follows we present low-complexity techniques in estimating prediction coefficients and correlation. We also compare the encoder complexity of the proposed system with two competing techniques, namely those based on inter-band prediction and 3D wavelets.

3.1. Estimation of predictor coefficients and correlation

The encoder can determine a rough level of correlation after it estimates α and β by computing an estimate of the residual energy after prediction. If this energy is above a certain threshold, the spectral band can be coded in intra-mode (i.e., independently of other bands); with the coding mode reverting to DSC mode when the residual energy goes under the threshold. For example, Band 162 in Fig. 1 can be coded in intra-mode. Note that in real data sets, we have considered a majority of bands can be coded using DSC (e.g., 95% of bands in the Cuprite data set we use in our experiments).

3.1.1. Estimation of predictor coefficients

As discussed earlier, we use a linear predictor $X' = \alpha X + \beta$ to generate side information for Y. The least-squares technique can be used to calculate
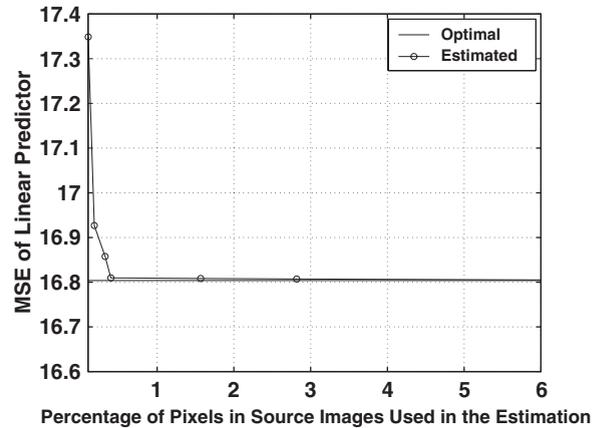

Fig. 6. MSE under first-order linear predictor for a typical spectral band.

α and β. In order to reduce the complexity (and data exchange requirements) of this process, we first down-sample the spectral bands and use only pixels in the down-sampled bands for estimation. As shown in Fig. 6, with only 0.32% of pixels, the resulting predictor can achieve a prediction mean-square error (MSE) within 0.05 of that of the optimal predictor (i.e., that computed using all pixels in X and Y). By using only a small fraction of data we reduce data exchange and computation in the least-squares calculation, without compromising the performance of the predictor (or its impact on the crossover probability estimation).

The overhead due to downsampling the data is usually negligible, as downsampling can be accomplished by incrementing the access position in data memory by a constant, and nowadays many CPU/DSPs have build-in hardware to support this operation and incur negligible overhead.

3.1.2. Estimation of crossover probability

We now consider estimation of the crossover probabilities at the encoder. These are needed to select an appropriate Slepian–Wolf coding rate at the encoder and to initialize the SPA at the decoder. To achieve low-cost estimation we propose that only a small portion of bit-plane data (generated by set partitioning) be exchanged between spectral bands. Note that, since set partitioning can be considered as a scrambling process on the ordering of coefficients, estimates of crossover probability after set partitioning tend to be reliable. We use the upper bound of the 95% statistical confidence interval as our estimate. Specifically, the upper

bound of the $(1 - \omega) \times 100\%$ confidence interval for a population proportion is given by [21]

$$\hat{p}_i = \frac{s_i}{n_i} + z_{\omega/2}\sqrt{p_i(1 - p_i)/n_i}$$

$$\approx \frac{s_i}{n_i} + z_{\omega/2}\sqrt{\frac{s_i}{n_i}\left(1 - \frac{s_i}{n_i}\right)/n_i}. \qquad (1)$$

Here $\hat{p}_i$ is the estimate of the crossover probability of bit-plane pair (b_i^x, b_i^y), n_i is the number of samples exchanged in estimating p_i, s_i is the number of exchanged samples for which crossover occurs, and $z_{\omega/2}$ is a constant that depends on the chosen confidence interval, e.g., $z_{\omega/2} = 1.96$ when we use a 95% confidence interval. Note that we choose the upper bound as the estimator to minimize the risk of decoding failure, at the expense of some encoding rate penalty. Statistically, with this estimation, we are $(1 - \omega) \times 100\%$ confident that the true crossover probability p_i is within $s_i/n_i \pm z_{\omega/2}\sqrt{p_i(1 - p_i)/n_i}$. Hence the estimation error, $\Delta p_i = \hat{p}_i - p_i$, is bounded by $0 \leqslant \Delta p_i \leqslant 2z_{\omega/2}\sqrt{p_i(1 - p_i)/n_i}$ with probability $1 - \omega$. In addition, it can be shown that (refer to the Appendix for details):

$$Pr(\Delta p_i < 0) = \omega/2,$$

$$Pr\left(\Delta p_i > 2z_{\omega/2}\sqrt{p_i(1 - p_i)/n_i}\right) = \omega/2,$$

which allows us to bound in a systematic way the probability of decoding error and the probability of incurring a large encoding rate penalty. Since the estimation process consists of simply counting of occurrences of crossovers in small portions of two bit-planes, the overall estimation overhead is small.

As an example of the accuracy of crossover probability estimation using this low-complexity technique, Fig. 7 shows a typical estimation result using different percentages of data from a bit-plane. As an example, with 5% of bits exchanged the crossover probability estimate is within 0.003 of the actual crossover probability. Since we choose the compression rate to leave a margin of about 0.05 bits over the Slepian–Wolf limit (as estimated by $H(\hat{p}_i)$, since we assume the source model as in [9]), this estimation accuracy is sufficient. In addition, we also test this technique in our coding performance experiments (details in Section 4). There we use around 10% of data in a bit-plane for correlation estimation and our experimental results show that the estimates are accurate enough that no decoding errors occur.
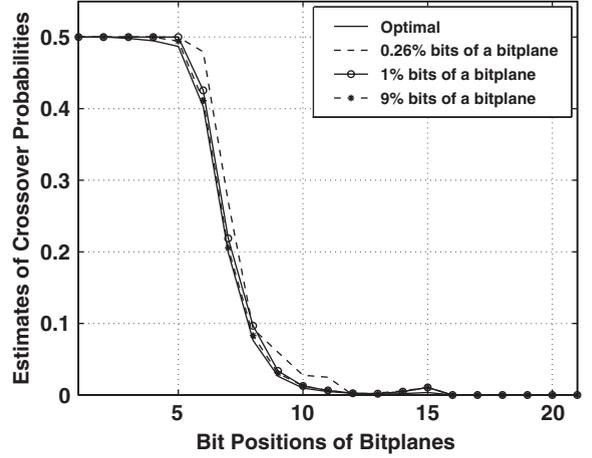

Fig. 7. Example of estimation of crossover probability. Note that for all bit-planes it is possible to achieve an accurate estimate using a small percentage of data in a bit-plane.

3.2. Encoder complexity comparison

In this section we compare the encoder complexity of our proposed scheme to that of inter-band prediction and 3D wavelet approaches.

3.2.1. Comparison with inter-band prediction

Inter-band prediction approaches need to generate exact copies of the decoded bands at the encoder, so that the encoder needs to perform decoding as well. To encode the current band Y using neighboring band X for prediction, the inter-band encoder requires following steps (Fig. 8):

(I-1)   Estimation of predictor coefficients α^* and β^* (in this case approximate techniques could also be used as long as the chosen parameters are communicated to the decoder).

(I-2)   Application of the prediction coefficients to obtain $\hat{X}' = \alpha^*\hat{X} + \beta^*$.

(I-3)   Computation of $Y - \hat{X}'$ to generate the residue.

(I-4)   Transformation of residue using the wavelet transform.

(I-5)   Set partitioning on the wavelet coefficients of residue. Output bitstream.

(I-6)   Inverse set partitioning.

(I-7)   Inverse transformation.

(I-8)   Adding $\hat{X}'$ to the output of inverse transformation to generate $\hat{Y}$.
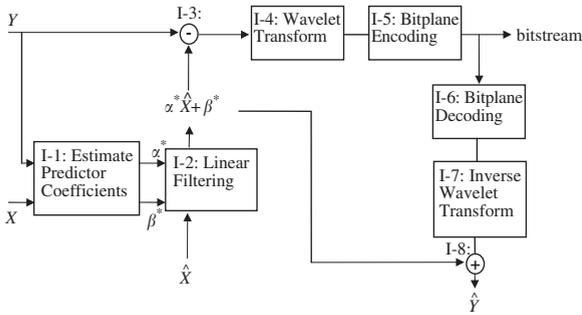
Fig. 8. Encoding using inter-band prediction.

Comparing the encoding steps of our proposed scheme (Fig. 4) with inter-band prediction approach (Fig. 8), we can make the following observations:

(i)   Both schemes need to compute the wavelet transform and perform bit-plane encoding of the current band Y (Steps (E-3) and (E-4) in Fig. 4, Steps (I-4) and (I-5) in Fig. 8).

(ii) The inter-band prediction approach has to perform an inverse wavelet transform (I-7). In our proposed scheme, we need wavelet coefficients of the linear predictor X' for correlation estimation. However, forward transformation is not necessary here since wavelet coefficients of X have been computed during the compression of previous band, and we can compute wavelet coefficients of X' simply by

$$T(X') = T(\alpha X + \beta)$$
$$= \alpha T(X) + \beta T(\mathbf{1}), \qquad (2)$$

where T denotes wavelet transformation, and $\mathbf{1}$ is a vector of ones. We pre-compute $T(\mathbf{1})$ and use it for all bands.

(iii) The inter-band prediction approach needs to perform bit-plane decoding (I-6).[3] In our proposed system we apply the significance tree of Y to the wavelet coefficients of X' to extract bit-planes (E-5), for crossover probability estimation. Note that in (E-5) we merely extract coefficients according to the significance tree of Y, and no significance

---

[3]Miguel et al.[3] has proposed using only full bit-planes to form the predictor. This could avoid bit-plane decoding at the encoder, but leads to performance degradation. In the general case when one wants to truncate at the middle of bit-plane, decoding of the significance bits is necessary to determine the order of the wavelet coefficients.

test on partition is required, so this is similar to (I-6) in terms of complexity. We would like to emphasize that the complexity of our system can be further reduced by avoiding bit-plane extraction, since there are low-complexity alternatives for correlation estimation. For example, our work in [22] has presented a technique in estimating crossover probability by first estimating the probability density function (pdf) of the wavelet coefficients, and then computing crossover probabilities analytically. Efficient pdf estimation techniques, such as those proposed in [23] for generalized Gaussian distributions, can be used and no bit-plane extraction will then be required.

(iv) The inter-band prediction approach requires subtracting the predictor from the current band to compute the residue (I-3), and then adding back the predictor to the reconstructed residue (I-8). Since the subtraction/addition has to be performed on every pixel, the complexity here is of the order of the amount of data in one band. On the other hand, our proposed scheme needs only a small portion of data to estimate crossover probabilities (E-6). Also the complexity of generating syndrome (E-8) is linear (since the parity-check matrix is sparse), and is of the order of the number of bit-planes we need to encode, which is usually small since in most lossy compression applications only high significance bit-planes are transmitted.

Based on the above comparisons, we conclude that our scheme requires lower encoding complexity than inter-band prediction approaches.

3.2.2. Comparison with 3D wavelet approaches

3D wavelet approaches operate on multiple spectral bands at the same time. This usually incurs substantial external memory access overheads in storing intermediate results. For example, using 3D wavelet approaches, 3D wavelet coefficients need to be computed first, followed by set partitioning of the 3D wavelet coefficients. Since the internal memory may not be able to accommodate several bands of 3D wavelet transform coefficients, they need to be transferred back and forth between

external and internal memory. In contrast, our proposed scheme operates on each single spectral band independently once the inter-band correlation has been estimated, and wavelet transformation and bit-plane encoding of a single spectral band can be completed entirely in internal memory without incurring external memory access for storing intermediate data. Hence the data access overheads in our scheme are much smaller than those involved in a 3D wavelet approach.

4. Implementation and experimental results

We have implemented SW-SPIHT and applied it to 16 bit hyperspectral images. The SPA we implemented for SW-SPIHT is based on the algorithm in Section (III-A) of [24] and similar notations are also used here. Consider the bipartite graph derived from the parity-check matrix, where each column of the matrix corresponds to a variable node in the graph and each row of the matrix corresponds to a check node in the graph. Let v_l be a variable node corresponding to the lth bit, and w_m be the check node corresponding to the mth check.

We made two changes to the algorithm in [24] in order to adapt it to decoding LDPC codes using side information. For details on these two changes, and further references on alternative sum–product decoding algorithms, we refer to the algorithm proposed in [25], for decoding correlated bit-planes using this belief propagation principle.

First, the initialization step makes use of a side-information bit-plane. Let q_{ml}^0 be the *a posteriori* probability of variable node v_l after an assignment of 0 which satisfies check node w_m. The a posteriori probabilities of variable nodes are initialized as follows:

$$(q_{ml}^0, q_{ml}^1) = (p^0, 1 - p^0), \quad 1 \leqslant l \leqslant n,$$

where n is the total number of check nodes, $p^0 = Pr(b_i^y(l) = 0|x_l)$, for these check nodes (w_m) which have an edge in the bipartite graph to variable node v_l. x_l is a binary scalar from the side-information bit-plane. Note that $Pr(b_i^y(l) = 0|b_i^x(l))$ equals to the crossover probability p_i if $b_i^x(l) \neq 0$.

The second change is on the check-node update step, in which we introduced a new local kernel function to force the search of the most probable codeword in a designated bin specified by the syndrome. Instead, the standard LDPC SPA searches the most probable codeword in the bin corresponding to the syndrome with all zero bits.

In our experiments we use data sets originally comprising 224 spectral bands, each of size 614×512 pixels. Due to constraints of the implementation of the codecs, in the experiments we compress 512×512 pixels in each band, and in total 192 bands starting from band number 33. Experimental results use SNR and PSNR for the comparison on individual frames and multiband SNR (MSNR) and multiband peak SNR (MPSNR) for the whole spectrum. These quantities are defined as follows:

$$\text{MSE} = E[(x - \hat{x})^2],$$

$$\text{SNR} = 10 \log_{10}\left(\frac{E[x^2]}{\text{MSE}}\right),$$

$$\text{PSNR} = 10 \log_{10}\left(\frac{(65535)^2}{\text{MSE}}\right),$$

where $E(.)$ is the expectation operator over pixels from an image band. x is the 16-bit value of a source pixel and $\hat{x}$ is the 16-bit value of reconstructed pixel of x. Also,

$$\text{MSNR} = 10 \log_{10}\left(\frac{E[x^2]}{\text{MSE}}\right),$$

$$\text{MPSNR} = 10 \log_{10}\left(\frac{(65535)^2}{\text{MSE}}\right),$$

where now $E(.)$ is the expectation operator over pixels from *all* spectral bands. The rates for individual image band is bits per pixel (bpp) and those for the whole spectrum are in bits per pixel per band (bpppb).

The outline of this experimental study is given as follows: first, we provide a comparison in terms of rate-distortion performance between SW-SPIHT and predictive 3D-SPIHT. Second, we compare SW-SPIHT with predictive 2D-SPIHT. In these experiments, we use different scenes and sites from the NASA AVIRIS data set including Cuprite Radiance (SC01), Moffet Field Radiance (SC03) and Lunar Lake Reflectance (SC02). In each experiment, all 224 bands are compressed.

In order to describe these alternative codecs and our implementations of them, we need the following notations:

1. A denotes a general image band.
2. B_i denotes the ith image band from the spectrum.

3. **1** is the vector with all 1 elements. The dimension is set as the number of pixels used by the least-squares predictor.

4. $V(A)$ is the function to vectorize a fixed number of pixels from image band A.

5. For the predictor image bands A and source image band B_i, $\alpha(A, B_i)$ is the prediction slope coefficient and $\beta(A, B_i)$ is the prediction intercept coefficient.

6. $B_i'(A)$ denotes the band after regression using least-squares prediction, and the design matrix is given by $X = (\mathbf{1}, V(A))$. Recall that the least-squares coefficients are given as follows:

$$(\beta, \alpha)^\tau = (X^\tau X)^{-1} X V(B_i),$$

where τ is the transpose operator.

7. Regression residuals of the least-squares predictor of the ith frame can be computed as $B_i - B_i'$.

4.1. Rate-distortion comparison with 3D wavelet approaches

Before presenting the rate-distortion comparison of SW-SPIHT with a predictive variant of 3D-SPIHT [4], we briefly describe these codecs and how we implemented them.

We modify 3D-SPIHT to adjust the bands taking into account their correlation. Thus, instead of operating on the original bands, $(B_1, B_2, \ldots,)$ we apply the wavelet transform and encoding to a new set of bands, $(B_1', B_2', \ldots,)$, obtained as follows:

1. $B_1' = B_1$.
2. For all $i > 1$, $B_i' = \alpha(B_i, B_{i-1}')B_i + \beta(B_i, B_{i-1}')$, and $\alpha(B_i, B_{i-1}')$ and $\beta(B_i, B_{i-1}')$ are directly encoded into bitstream.

We use this predictive 3D-SPIHT approach so as to better "align" all spectral bands, so that wavelet transform can better exploit the inter-band correlation.

Similarly, as previously mentioned, in the SW-SPIHT codec, we take into account the prediction accuracy and when the estimated value for the residual MSE is high the standard SPIHT is used, i.e., the spectral band is coded in intra-mode. Also, individual bit-planes with an estimated crossover probability greater than a threshold are intra-coded. This information (intra-coded bands or bit-planes) is explicitly conveyed to the decoder.

Fig. 9 provides coding performance comparisons for the radiance data from the Cuprite and Moffet Field sites, and the reflectance data from the Lunar Lake site (log scale is used for the rate to facilitate the comparison at low bit-rates). To obtain the results of predictive 3D-SPIHT, we use an implementation of 3D-SPIHT [27,28] available in the public domain. It can be seen that SW-SPIHT performs competitively, with marginal gain over 3D-SPIHT at most rate regions. In addition, SW-SPIHT has moderate memory requirement for encoding. It should be noted that the performance of 3D-SPIHT can be improved by applying entropy coding (e.g., arithmetic coding) on the output bits. Similarly, we can improve our SW-SPIHT by applying entropy coding on the significance bits information. Also note that results for 3D-SPIHT without prediction (not included here) are close to predictive 3D-SPIHT with a marginal loss at low bit-rates.

It is well known that wavelet set-partitioning-based codecs can precisely control the bit-rate. In other words, the SNR can be kept at a required level when the bit-rate is allowed to change. However, this only holds for global SNR, and not necessarily for different parts of the encoded stream. In the case of 3D-SPIHT, the SNR of individual spectral bands can actually fluctuate significantly for a given target global SNR (variations of up to 5 dB are possible, see Fig. 10 for an example). Another salient feature of SW-SPIHT is that it allows targeting individual band SNRs, so that fluctuations across bands can be kept very small (e.g., within 1 dB). Note that
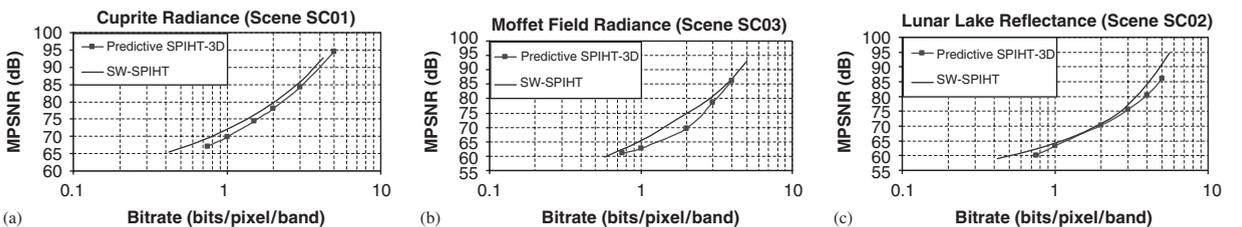

Fig. 9. Rate-distortion curves of SW-SPIHT and predictive 3D-SPIHT: (a) Cuprite; (b) Moffet Field; (c) Lunar Lake.

these variations are undesirable, as they could destroy the spectral signatures that are of primary interest in analysis of hyperspectral imagery. Refer to Section 5 for an example of how SW-SPIHT is better at preserving these spectral signatures.

4.2. Rate-distortion comparison with 2D wavelet approaches

We have also implemented two other 2D wavelet-based codecs. The first is the standard 2D-SPIHT codec that operates independently on all spectral bands, without crossband prediction. The second is the predictive 2D-SPIHT codec, which operates as follows:

1. The first image band B_1 is encoded as is.
2. $\hat{B}_{i-1}$, reconstruction of image band B_{i-1}, is used to obtain a predictor for B_i, B'_i.
3. 2D-SPIHT codec is applied to $B_i - B'_i$ for all $i > 1$; if the residual energy is above a certain threshold then B_i is enclosed directly.
4. Prediction coefficients $\alpha(\hat{B}_{i-1}, B_i)$ and $\beta(\hat{B}_{i-1}, B_i)$ are sent as overhead.

Note that the predictor used in 2D-SPIHT is the preceding image band, and this is different from the predictor used in the predictive 3D-SPIHT codec.

Fig. 11 provides comparisons based on the radiance data from the Cuprite and Moffet Field sites, and the reflectance data from the Lunar Lake site. For Cuprite site, SW-SPIHT achieves marginal gain at middle range bit-rates, but suffers marginal loss at high bit-rates. The coding performance of predictive 2D-SPIHT improves at high bit-rates thanks to the better quality reconstruction used as predictor. For Moffet Field and Lunar Lake sites, SW-SPIHT achieves marginal gain consistently,
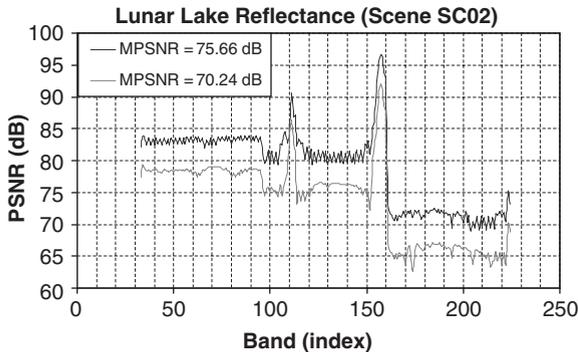

Fig. 10. Inter-band SNR fluctuation under 3D-SPIHT.

demonstrating competitive rate-distortion performance.

Fig. 12 compares the performance of SPIHT and SW-SPIHT on individual image bands. We selected three pairs of bands from different spectral regions where the levels of correlations are different as also shown in Fig. 1. We did not select bands in spectral regions where the predictor sees large surges on mean-square residuals and these bands have low correlation and intra-coding on these bands is used instead. In Fig. 12, SW-SPIHT outperforms SPIHT significantly, with up to 5dB gain at some rate regions. There are some variations on the PSNR gain due to variations of the energy among these images and correlations between images in these pairs.

5. Discussion

In the previous section, we have demonstrated that SW-SPIHT performs competitively comparing to other set-partitioning-based codecs. In this section, we will shed some light on why Slepian–Wolf bit-plane coding after set partitioning is better than Slepian–Wolf coding on raw bit-planes without set partitioning. We will also present results in this section to illustrate the improved preservation of spectral signature under SW-SPIHT.

5.1. Slepian–Wolf coding of bit-planes

In our proposed method we have applied Slepian–Wolf to bit-planes comprising only sign and refinement bits. To illustrate why this is better than applying the same coding technique to raw bit-planes, we compared the rates achieved by Slepian–Wolf coding of set-partitioning output and that of raw bit-planes. To estimate the rates, we measured the crossover probabilities between bit-planes for these two cases and computed their respective coding rate lower bounds. For the case with set partitioning, we also take into account the significance bits when computing the rates. The same least-squares predictor is used for both cases.

Table 1 compares these two approaches in a case where the MSE of prediction residuals is 14.87. Here, b_1 and t_1 are the Slepian–Wolf compressed bit-plane width and uncompressed bit-plane width, respectively, when set partitioning is performed before encoding. b_2 and t_2 are the Slepian–Wolf compressed bit-plane width and uncompressed bit-plane width, respectively, when the raw bit-planes
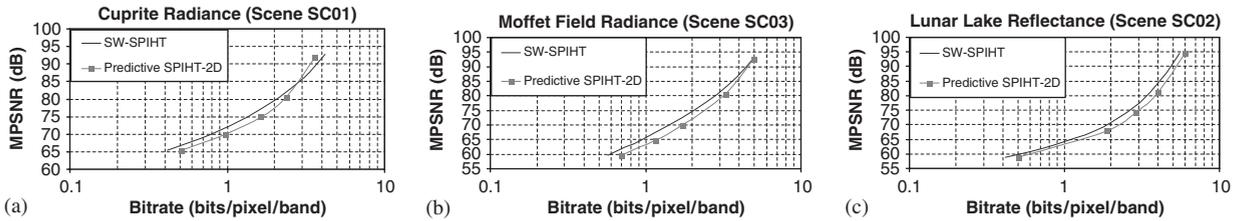
Fig. 11. Rate-distortion curves of SW-SPIHT and predictive 2D-SPIHT: (a) Cuprite; (b) Moffet Field; (c) Lunar Lake.
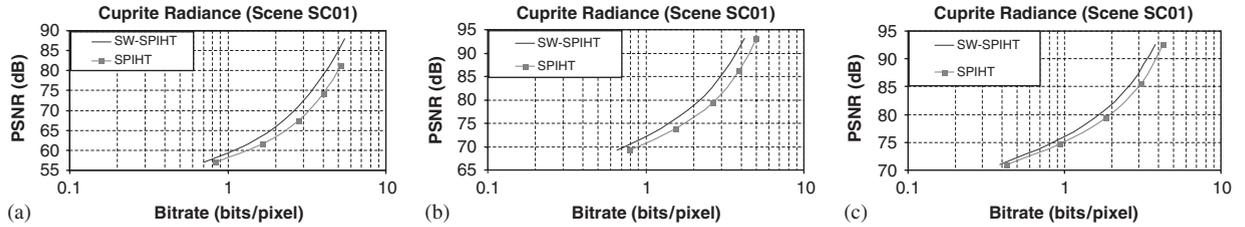

Fig. 12. Rate-distortion curves of SW-SPIHT and SPIHT (site: Cuprite, View: SC01): (a) band 40; (b) band 133; (c) band 190.

Table 1
Set-partitioning gain on Cuprite radiance spectrum

Significance level	Set-partitioning outputs				Raw bit-planes		
	b_1	t_1	Significance	r_1	b_2	t_2	r_2
15	108	108	1792	0.007	148675	524288	0.567
14	364	364	3476	0.015	174908	786432	0.667
13	620	620	5012	0.021	201142	1048576	0.767
12	880	880	6548	0.028	227376	1310720	0.867
11	1182	1182	8125	0.036	253610	1572864	0.967
10	1744	1744	10410	0.046	279971	1835008	1.068
9	3321	3321	16229	0.075	306660	2097152	1.170
8	8039	8039	32729	0.156	334642	2359296	1.277
7	10412	21126	73164	0.319	367315	2621440	1.401
6	17960	53099	153660	0.655	416002	2883584	1.587
5	40267	120426	285472	1.243	507542	3145728	1.936
4	101193	240667	451763	2.109	678079	3407872	2.587
3	229661	416565	595444	3.148	938715	3670016	3.581
2	425301	632331	683658	4.230	1200859	3932160	4.581

are encoded directly. To calculate the coding rates achieved by Slepian–Wolf coding of set-partitioning outputs, r_1, and that of raw bit-planes, r_2, we first calculate the conditional entropy between side-information bit-planes and the coding bit-planes, and it is multiplied by the number of bits of the bit-plane to obtain the Slepian–Wolf lower bound. In case of set-partitioning outputs, we also add the number of significance bit, which is uncompressed. We then divide the results by the number of pixel in

an image band to obtain the rates. Comparing the rates of these two approaches, the case with set partitioning outperforms the case without set partitioning in most bit-planes, with performance very close at low bit-rates.

Fig. 13(a)–(c) shows the rate comparison plots from Cuprite, Moffet Field and Lunar Lake sites, respectively, at varying MSE of predicted residuals. The gain from set-partitioning process is persistent in these cases.
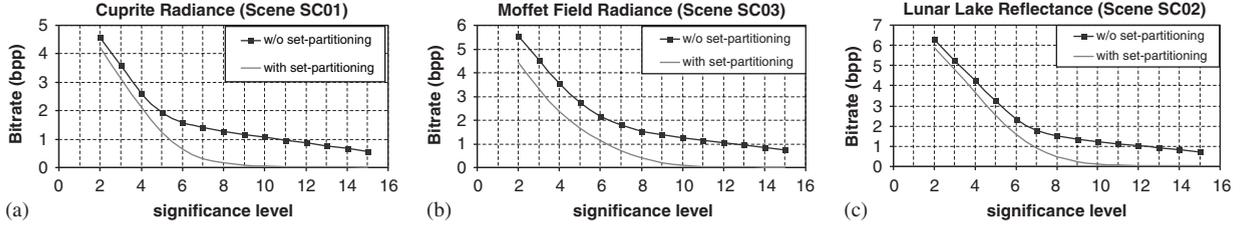
Fig. 13. Set-partitioning gain: (a) Cuprite at 14.87 MSE; (b) Moffet Field at 151.96 MSE; (c) Lunar Lake at 250.54 MSE.
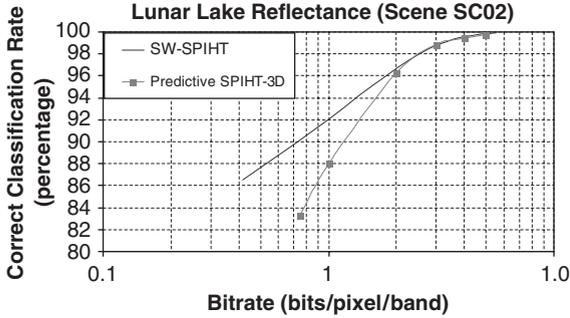

Fig. 14. Classification performance (site: Lunar Lake, View: SC02).

5.2. Preservation of spectral signature

As mentioned earlier, SW-SPIHT allows encoding with very consistent quality across bands, a property that cannot be guaranteed with 3D-SPIHT. To illustrate the potential advantages of SW-SPIHT in terms of signature preservation, we have also assessed its performance in a remote sensing classification application. We tested our system with Spectral Angle Mapper (SAM) algorithm [26], which is a well-known algorithm designed to measure the similarity between the unknown test spectra and the reference spectra. Similar to the setup in [4], we assume that the classification results of the original image are correct, and measure the number of pixels of the reconstructed image which have the same classification results as the original image pixels. Fig. 14 depicts the comparison of classification performance. As shown in the figure our proposed approach outperforms 3D-SPIHT in general. This is because our approach can keep the variation of SNR small across bands. As a result, spectral signatures can be better preserved.

6. Conclusions

In this paper, we have demonstrated a viable approach for compression of hyperspectral imagery. A novel scheme called SW-SPIHT is proposed. Our scheme has low computation and memory requirements for encoding, which are critical for hyperspectral imagery applications. In addition, encoding can proceed in parallel once the correlation statistics are estimated. This enables efficient parallel hardware implementations. Also we have shown that estimation of correlation statistics requires only limited data exchange across bands. As for coding performance, we have compared our scheme with several existing techniques including 3D-SPIHT, predictive 2D-SPIHT and SPIHT. Experimental results show that our scheme can achieve competitive coding efficiency. Furthermore, our scheme can preserve spectral signatures and obtain good classification performance.

Acknowledgment

Appendix

Here, we justify the two equations in Section 3.1.2. We let the crossover probability estimator be the upper bound of the $(1 - \omega) \times 100\%$ confidence interval for a population proportion, i.e.,

$$\hat{p}_i = \frac{s_i}{n_i} + z_{\omega/2}\sqrt{p_i(1 - p_i)/n_i}$$

$$\approx \frac{s_i}{n_i} + z_{\omega/2}\sqrt{\frac{s_i}{n_i}\left(1 - \frac{s_i}{n_i}\right)/n_i}.$$

Let $m = z_{\omega/2}\sqrt{p_i(1-p_i)/n_i}$. By the definition of confidence interval, we have

$$Pr\left(\frac{s_i}{n_i} - m \leqslant p_i \leqslant \frac{s_i}{n_i} + m\right) = 1 - \omega.$$

Equivalently,

$$Pr\left(p_i - m \leqslant \frac{s_i}{n_i} \leqslant p_i + m\right) = 1 - \omega.$$

By this and the fact that si/ni can be approximated by a normal density with mean p_i and variance $p_i(1-p_i)/n_i$, we have

$$Pr\left(\frac{s_i}{n_i} < p_i - m\right) = \omega/2,$$

$$Pr\left(\frac{s_i}{n_i} + m - p_i < 0\right) = \omega/2,$$

$$Pr(\hat{p}_i - p_i < 0) = \omega/2$$

and

$$Pr\left(\frac{s_i}{n_i} > p_i + m\right) = \omega/2,$$

$$Pr\left(\frac{s_i}{n_i} + m - p_i > 2m\right) = \omega/2,$$

$$Pr(\hat{p}_i - p_i > 2m) = \omega/2.$$

From these equations, the probability of decoding error and probability of large encoding rate penalty can be estimated.

References

[1] C. Tang, N.-M. Cheung, A. Ortega, C.S. Raghavendra, Efficient inter-band prediction and wavelet based compression for hyperspectral imagery: a distributed source coding approach, in: Proceedings of Data Compression Conference, DCC 2005, March 2005, pp. 437–446.

[2] D. Landgrebe, Hyperspectral image data analysis, IEEE Signal Process. Magazine 19 (January 2002).

[3] A.C. Miguel, A.R. Askew, A. Chang, S. Hauck, R.E. Ladner, Reduced complexity wavelet-based predictive coding of hyperspectral images for FPGA implementation, in: Proceedings Data Compression Conference, 2004.

[4] X. Tang, W.A. Pearlman, J.W. Modestino, Hyperspectral image compression using three-dimensional wavelet coding: a lossy-to-lossless solution, IEEE Trans. Geosci. Remote Sensing, submitted for publication.

[5] D. Slepian, J.K. Wolf, Noiseless coding of correlated information sources, IEEE Trans. Inform. Theory (July 1973) 471–480.

[6] S. Pradhan, K. Ramchandran, Distributed source coding using syndromes (DISCUS), IEEE Trans. Inform. Theory 49 (March 2003) 626–643.

[7] J. Garcia-Frias, Y. Zhao, Compression of correlated binary sources using turbo codes, IEEE Comm. Lett. (October 2001) 417–419.

[8] A. Aaron, B. Girod, Compression with side information using turbo codes, in: Proceedings of Data Compression Conference, 2002.

[9] A.D. Liveris, Z. Xiong, C.N. Georghiades, Distributed compression of binary sources with side information at the decoder using LDPC codes, IEEE Commun. Lett. (October 2002) 440–442.

[10] Z. Xiong, A. Liveris, S. Cheng, Distributed source coding for sensor networks, IEEE Signal Process. Magazine (September 2004) 80–94.

[11] S. Pradhan, J. Kusuma, K. Ramchandran, Distributed compression in a dense microsensor network, IEEE Signal Process. Magazine (March 2002) 51–60.

[12] B. Girod, A. Aaron, S. Rane, D. Rebollo-Monedero, Distributed video coding, Proceedings of the IEEE, Special Issue on Advances in Video Coding and Delivery, vol. 93, January 2005, pp. 71–83.

[13] R. Puri, K. Ramchandran, PRISM: a new robust video coding architecture based on distributed compression principles, in: Proceedings of the Allerton Conference on Communications, Control, and Computing, October 2002.

[14] A. Said, W.A. Pearlman, A new, fast, and efficient image codec using set partitioning in hierarchical trees, IEEE Trans. Circuits Systems for Video Technol. (June 1996) 243–250.

[15] A.D. Liveris, C.-F. Lan, K. Narayanan, Z. Xiong, C.N. Georghiades, Slepian–Wolf coding of three binary sources using LDPC codes, in: Proceedings of the International Symposium on Turbo Codes and Related Topics, September 2003.

[16] H. Wang, N.-M. Cheung, A. Ortega, WZS: Wyner–Ziv scalable predictive video coding, in: Proceedings of Picture Coding Symposium, 2004.

[17] H. Wang, N.-M. Cheung, A. Ortega, A framework for adaptive scalable video coding using Wyner–Ziv techniques, EURASIP J. Appl. Signal Process., accepted for publication.

[18] M. Pedram, E.J.M. Rabaey, Power Aware Design Methodologies, Kluwer Academic Publishers, Dordrecht, 2002.

[19] A.R. Calderbank, I. Daubechies, W. Sweldens, D. Yeo, Wavelet transforms that map integers to integers, Appl. Comp. Harmonic Anal. 5 (1998) 332–369.

[20] N.T. Slingerland, A.J. Smith, Cache performance for multimedia applications, in: IEEE International Conference on Supercomputing, 2001.

[21] W. Mendenhall, T. Sincich, Statistics for Engineering and the Sciences, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[22] N.-M. Cheung, H. Wang, A. Ortega, Correlation estimation for distributed source coding under information exchange constraints, in: IEEE International Conference on Image Processing, 2005.

[23] M.N. Do, M. Vetterli, Wavelet-based texture retrieval using generalized Gaussian density and Kullback–Leibler distance, IEEE Trans. Image Process. 11(2) (2002) 146–158.

[24] D.J.C. MacKay, Good error-correcting codes based on very sparse matrices, IEEE Trans. Inform. Theory (March 1999) 399–431.

[25] C. Tang, C.S. Raghavendra, Bitplane coding for correlations exploitation in wireless sensor networks, in: Proceedings of the IEEE International Conference on Communications, Seoul, May 2005.

[26] J.E. Fowler, An open source software library for quantization, compression and coding, in: Proc. SPIE Applications of Digital Image Processing XXIII, August 2000, pp. 294–301.

[27] J.E. Fowler, QccPack, Quantization, compression, coding library and utilities, http://qccpack.sourceforge.net.

[28] R.N. Clark, G. Swayze, J. Boardman, F. Kruse, Comparison of three methods for materials identification and mapping with imaging spectroscopy, in: JPL AVIRIS Airborne Geoscience Workshop, 1993.