LOW-COMPLEXITY AND ERROR-RESILIENT HYPERSPECTRAL IMAGE COMPRESSION BASED ON DISTRIBUTED SOURCE CODING

A. Abrardo[a], M. Barni[a], A. Bertoli[c], A. Garzelli[a], E. Magli[b], F. Nencini[a], B. Penna[b], R. Vitulli[d]

[a]Dip. di Ingegneria dell'Informazione - Università di Siena
Via Roma, 56 - 53100 Siena - Italy
(abrardo,barni,garzelli,nencini)@dii.unisi.it

[b]Dipartimento di Elettronica - Politecnico di Torino
Corso Duca degli Abruzzi 24 - 10129 Torino - Italy
enrico.magli(barbara.penna)@polito.it

[c]Carlo Gavazzi Space S.p.A.
Via Gallarate 150 - 20151 Milano - Italy
abertoli@cgspace.it

[d]On-Board Payload Data Processing Section
European Space Agency - ESTEC TEC/EDP
Noordwijk, The Netherlands
Raffaele.Vitulli@esa.int

ABSTRACT

In this paper we propose a lossless compression algorithm for hyperspectral images based on distributed source coding; this algorithm represents a significant improvement over our prior work on the same topic, and has been developed during a project funded by ESA-ESTEC. In particular, the algorithm achieves good compression performance with very low complexity; moreover, it also features a very good degree of error resilience.

These features are obtained taking inspiration from distributed source coding, and particularly employing coset codes and CRC-based decoding. As the CRC can be used to decode blocks using a reference different from that used to compress the image, this yields error resilience. In particular, if a block is lost, decoding using the closest collocated block in the second previous band is successful about 70% of the times.

Keywords: Hyperspectral compression; lossless compression; distributed source coding; error resilience.

1. INTRODUCTION

Compression of multispectral and hyperspectral images has recently received a lot of attention. New sensors are generating increasing amounts of data, especially in the spectral dimension, as scenes are imaged at a very fine wavelength resolution. This is particularly useful in terms of potential applications, as spectral features allow to extract important information from the data. However, it also makes the size of the acquired images extremely high. Since many sensors, especially spaceborne ones, cannot store all the data but need to transmit them to a ground station, there is a problem of reducing the data size in order to download all the acquired data.

Image compression techniques can be employed to mitigate this problem, allowing to transmit more scenes in the same amount of time. Several types of compression are possible. In lossless compression, the reconstructed image is identical to the original. In near-lossless compression, the maximum absolute difference between the reconstructed and original image does not exceed a user-defined value. In lossy compression, the reconstructed image is as similar as possible to the original "on average", i.e., typically in mean-squared error sense, given a target bit-rate. Lossless compression is highly desired to preserve all the information contained in the image; unfortunately, the best algorithms provide limited compression ratios. For example,[1,2] achieve a compression ratio of about 3.5:1 on 16 bpp AVIRIS data. Near-lossless and lossless techniques yield larger size reductions, at the expense of some information loss. E.g., in [3] it is shown that bit-rates of 0.5 and 0.1 bpp can be achieved with little or no impact on image classification performance.

While compression is becoming an increasingly important part of on-board processing, it should be noted that its role is not only that of reducing the data size. In particular, the following functionalities are of interest.

- **Compression with low encoder complexity** Algorithms should reduce the data size as much as possible; this is the traditional and core functionality of any image compression scheme. For on-board compression, it is of paramount importance that the encoder has *low-complexity*, in order to match the low computational capabilities and the high sensor data rate.
- **Error-resilience** Algorithms should be capable of dealing with "errors". This requirement is important because of the way the data are processed and transmitted to the ground segment. Until a few years ago it was customary to employ radiation-hardened digital signal processors to perform on-board processing, including compression. Radiation-hardened components have the same functionality of an equivalent standard processor, but they are designed to be insensitive to ionization. Such processors are extremely expensive to design and manufacture; as a result, the available processors do not have state-of-the-art computation capabilities. This has led to a recent trend of employing off-the-shelf processors, as in.[4] Off-the-shelf processors are prone to the effect of radiations; e.g., a typical effect is to occasionally flip some bits in the on-board memory. Moreover, when the data are transmitted to the ground station, there is a small possibility that the data are corrupted. As a consequence, modern compression algorithms should be able to manage the event of errors in the data. Traditional compression algorithms such as those in[2,5] will break down completely upon a single bit error in the data, preventing from decoding the remainder of the scene after the error. Conversely, it is required that compression algorithms limit the scope of errors to a small part of the data, or are able to correct errors to some extent. The importance of error-resilience is also witnesses by its inclusion as a requirement for the next generation of lossless and lossy multispectral and hyperspectral compression standards developed by the CCSDS (consultative committee for space data systems).

In this paper we propose a lossless compression algorithm that provides the two functionalities described above, namely low-complexity compression and error-resilience. The algorithm is inspired by recent ideas in the field of distributed source coding (DSC).[6] It builds on our previous algorithm presented in,[7,8] improving it in many aspects, and particularly significantly boosting the compression performance, decreasing complexity, and providing error-resilience. This algorithm has been proposed at the spring 2008 meeting of the CCSDS multispectral and hyperspectral data compression group as candidate for standardization.

This paper is organized as follows. In Sect. 2 we provide some background on DSC. In Sect. 3 we describe the proposed algorithm. In Sect. 4 we report experimental results, while in Sect. 5 we draw some conclusions.

2. BACKGROUND

In recent years, DSC has received an increasing attention from the signal processing community. DSC considers a situation in which two (or more) statistically dependent data sources must be encoded by separate encoders that are not allowed to talk to each other. Performing separate lossless compression may seem less efficient than joint encoding. However, DSC theory proves that, under certain assumptions, separate encoding is optimal, provided that the sources are decoded jointly.[9] For example, with two sources it is possible to perform "standard" encoding of the first source Y (called *side information*) at a rate equal to its entropy, and "conditional" encoding of the second source X at a rate lower than its entropy, with no information about the first source available at the second encoder.

DSC theory also encompasses lossy compression[10]; it has been shown that, under certain conditions, there is no performance loss in using DSC,[10,11] and that possible losses are bounded below 0.5 bit per sample (bps) for quadratic distortion metric.[12] In practice, lossy DSC is typically implemented using a quantizer followed by lossless DSC. Lossless and lossy DSC have several potential applications, e.g., coding for non co-located sources such as sensor networks, distributed video coding,[13] layered video coding,[14,15] and remote sensing image compression,[7,16] just to mention a few.

For remote sensing image compression, which is addressed in this paper, the idea is to consider each band X_i of a multispectral or hyperspectral image as an individual source. Adjacent bands are correlated, and this raises the problem of exploiting their correlation. Classical algorithms do so by employing interband prediction, i.e., performing modeling at the encoder. On the contrary, DSC techniques perform little or no modeling at the encoder, achieving lower encoder complexity and error-resilience, and achieving ideally the same bit-rate as a classical joint encoder, with vanishing probability of not being able to successfully decode the image. Indeed, in DSC it is the decoder that is burdened with the modeling task; however, in remote sensing applications this is not an issue as the computation constraints at the ground segment are not nearly as tight as on-board.

In particular, DSC is applied to remote sensing images as in,[7,8] by encoding each band X_i using DSC techniques, and using the previous band X_{i-1}, which has already been decoded, as side information. Thus, the encoder compresses band X_i at rate approximately equal to $H(X_i|X_{i-1})$, and the decoder reconstructs X_i using X_{i-1} as side information. An example of this process will be shown later in this section.

Traditional entropy coding of an information source can be performed using one out of many available methods, the most popular being arithmetic coding and Huffman coding. "Conditional" (i.e., DSC) coders are typically implemented using channel codes, by representing the source using the syndrome or the parity bits of a suitable channel code of given rate. The syndrome identifies sets of codewords ("cosets") with maximum distance properties, so that decoding an ambiguous description of a source at a rate less than its entropy (given the side information) incurs minimum error probability. If the correlation between X and Y can be modeled as a "virtual" channel described as $X = Y + W$, with W an additive noise process, a good channel code for that transmission problem is also expected to be a good S-W source code.[11] In practice, trellis channel codes,[17] turbo codes[18] and low-density parity-check (LDPC) codes[19] have been used; distributed arithmetic codes have also been proposed.[20]

Since the codes above still have a relatively high encoder complexity, in this paper we employ a scalar coset code, which exhibits lower coding performance but significantly smaller complexity. We describe the scalar coset code referring to Fig. 1 by means of an example. We assume that the samples of each band are represented on three bits, assuming values from 0 to 7. We divide the eight values into four cosets, identified by triangle-up, square, circle and triangle-right. Each coset contains two values taken at maximum distance between each other. Compression operates as follows. Let us say for example that we want to encode the current sample X, which happens to have value equal to 2. If we did not use the previous band, we would need 3 bpp to specify all possible values of X. Instead, we label each coset and send to the decoder only the label of the coset that X belongs to. In this example we have four cosets, hence we need two bits; compared with the three bits of the previous case, the rate has been reduced by one third. The decoder receives the coset label for X, which tells her that the X is in $\{2, 6\}$. It employs the correlated side information, i.e., the co-located sample Y in the previous band, to disambiguate between these two values, by choosing for X the value that is closest to Y. For example, if $Y = 3$, then the decoder will reconstruct $X = 2$. Intuitively, if Y is correlated with X, the probability of error, i.e. that Y is closer to another element of the coset other than the true value of X, is vanishingly small.

This simple example allows to highlight a few important aspects of DSC. One of them is the relation between the number and size of the cosets. In the example above, instead of taking four cosets with two elements in each, we could have taken two cosets with four elements in each, transmitting only one bit as label for X. The optimal number of coset actually depende on how much correlated X and Y are. The larger the number of cosets, the larger the minimum distance between two elements of the same cosets. Therefore, if X and Y are only mildly correlated, we need a large minimum distance to make sure that the decoder will be able to disambiguate X, and hence we will need more bits for the label. Viceversa, if X and Y are very strongly correlated, we can create less cosets and use less bits. A typical choice is to take the number of cosets as a power of two, i.e., 2^K. This choice makes it easy to derive the label for any X, as the label are simply the K least significant bits of X.

Another important aspect regards error resilience. The reason why existing algorithms are not error resilient is that the predictor of the current sample X is a specific value that can be computed in a causal way only if all previous pixels of the image have been decoded without errors. With DSC, this is not the case. In fact, *any* side information Y that is sufficiently close to X will allow the decoder to decode correctly. Let us suppose that the co-located pixel $Y = 3$ in the

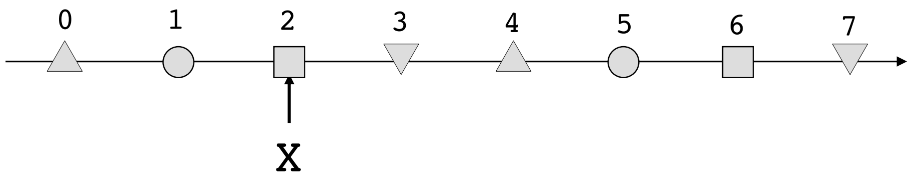

Figure 1. DSC using scalar coset codes.

previous band has been corrupted, and that wrong value $Y' = 1$ is available. As long as Y' is closer to X than to any other element of the same coset, the decoder will provide the correct value of X.

3. DESCRIPTION OF THE PROPOSED ALGORITHM

3.1. Encoding

The structure of the algorithm is described in two steps. In the first step, a modified version of the DSC algorithm in [7] is introduced, then some changes are introduced to provide error resilience.

Let $x_{m,n,i}$ denote the pixel of an hyperspectral image X in $m-th$ line, $n-th$ column, and $i-th$ band. Suppose that each band is partitioned into non-overlapping blocks of size 16×16, and that each pixel $x_{m,n,i}$ can be recovered from the pixel of previous band $X_{m,n,i-1}$ with a linear model

$$x_{m,n,i} = f(x_{m,n,i-1}) + v_{m,n} \tag{1}$$

where $v_{m,n}$ a noise process describing the correlation. The value of pixels in X is coded by retaining only the least significant bits (LSBs) of $x_{m,n,i}$, where the selected value of LSBs depends on the amount of correlation between the current and the previous band. In [7] the number of LSBs is fixed for all pixels within the block, and the MSBs are recovered on the basis of the side information $x_{m,n,i-1}$. The number of LSBs is selected in such a way as to guarantee that the decoder will be able to exactly reconstruct X.

To compress $x_{m,n,i}$, the following steps are applied.

1. In order to make $x_{m,n,i-1}$ as similar as possible to $x_{m,n,i}$ in a Minimum Mean-Squared Error sense, a Least-Squares estimator is computed as follows

$$\alpha_{[-1]} = \frac{\sum_{m,n}(x_{m,n,i-1} - \mu_{i-1}) \cdot (x_{m,n,i} - \mu_i)}{\sum_{m,n}(x_{m,n,i-1} - \mu_{i-1}) \cdot (x_{m,n,i-1} - \mu_{i-1})}, \tag{2}$$

   where μ_i and μ_{i-1} are the average value of corresponding blocks of bands i and $i-1$.

2. A quantized version of $\alpha_{[-1]}$ is generated with a uniform scalar quantizer with 200 levels starting from 0 to 1.99. The predicted values within the block will be

$$\widetilde{x}_{m,n,i} = \mu_i + \widehat{\alpha}_{[-1]}[x_{m,n,i-1} - \mu_{i-1}]; \tag{3}$$

3. The error vector can be calculated in each block as

$$e^{[-1]}_{m,n} = x_{m,n,k} - \widetilde{x}_{m,n,k} \tag{4}$$

   and the maximum error is used to set the number of LSBs to be retained as

$$k_{[-1]} = \left\lfloor \log_2(||e^{[-1]}_{m,n}||_\infty) \right\rfloor + 2; \tag{5}$$

4. The $k_{[-1]} - 1$ LSBs are transmitted and for the $k_{[-1]}$ least significant bit-plane the following map is computed

$$M_{[-1]} = \begin{cases} 0 & |e^{[-1]}_{m,n}| < 2^{k_{[-1]}-2} \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

where the maximum absolute error in the block is $2^{k_{[-1]}-1}$. The map $M_{[-1]}$, as well as the $k_{[-1]}$-th LSB of those pixe;s for which $M_{[-1]} = 1$, need to be written in the compressed file.

We expect that, on average, the error will be smaller than $2^{k_{[-1]}-2}$, so that most samples of $M_{[-1]}$ will be zero. This means that $M_{[-1]}$ is a highly compressible signal. To take advantage of this, we decide to transmit only the positions of the bits of $M_{[-1]}$ that are equal to one; these positions are coded using a differential Huffman coding scheme, and then written in the compressed file. In the decoding process, the pixel value of the current band will be recovered from $k_{[-1]} - 1$ LSBs when the prediction error is lower than $2^{k_{[-1]}-2}$, or from $k_{[-1]}$ LSBs when $M_{[-1]}$ is equal to 1. In particular, each pixel is decoded by taking the element of the coset that is closest to the side information in Euclidean sense.

5. For each block, the compressed bitstream contains also $k_{[-1]}$ value, as well as 32 parity-check bits obtained by applying a CRC code to the values of the pixels in the block. The CRC code must have good error detection properties, in order to not hinder the decoding process, and must be fast. For example, fast non-CRC error detection codes such as the Internet checksum are not adequate because of small minimum distance. We have used a 32-bit CRC with polynomial $d(x) = x^{32} + x^{31} + x^8 + 1$,,[21] whose complexity is about 5.75 instructions per input byte.

Error resilience is achieved noting the following. The main limitation of the proposed algorithm is that all parameters have been calculated in order to recover the current band with $x_{m,n,i-1}$ as side information. However, if $x_{m,n,i-1}$ has been corrupted by errors, it cannot be used as side information. This means that all co-located block in the subsequent bands would be lost, as the decoder would be unable to reconstruct them. In this paper we overcome this issue by using $x_{m,n,i-2}$ as an additional side information, and modifying the procedure described above in order to retrieve the current band from either $x_{m,n,i-1}$ or $x_{m,n,i-2}$, if there exists a sufficiently strong correlation.

First, $\alpha_{[-2]}$, $e^{[-2]}_{m,n}$, $k_{[-2]}$ and $M_{[-2]}$ are calculated between $x_{m,n,i}$ and $x_{m,n,i-2}$, repeating the same operations described above. Then, three different conditions are evaluated:

- $k_{[-2]} < k_{[-1]}$;
- $k_{[-2]} = k_{[-1]}$;
- $k_{[-2]} > k_{[-1]}$.

When $k_{[-2]}$ is lower than $k_{[-1]}$, $k_{[-2]}$ LSBs are enough to recover the current band values. Indeed, in this case the second previous band is more correlated with the current band than the previous band. On the contrary, the reconstruction of $x_{m,n,i}$ is not guaranteed if $k_{[-2]} > k_{[-1]}$. However, as band $i-2$ is typically close to band $i-1$, most of the times the decoder should be able to reconstruct the current band using $k_{[-2]}$ LSBs and $x_{m,n,i-2}$ as side information.

To make this possible, we define a new $M_{[-1]}$ map through the following equation

$$\widehat{M}_{[-1]} = M_{[-1]} | M_{[-2]}. \tag{7}$$

when $k_{[-1]} = k_{[-2]}$. The symbol "|" denotes the logical "OR" operator. The new map allows to recover $x_{m,n,i}$ from $x_{m,n,i-1}$ and $x_{m,n,i-2}$, because $k_{[-1]}$ LSBs are transmitted when $e^{[-1]}_{m,n}$ or $e^{[-2]}_{m,n}$ are grater than $2^{k_{[-1]}-2}$. In addition to $k_{[-1]-1}$ LSBs the new bitstream contains the compressed $\widehat{M}_{[-1]}$ errors positions and the corresponding $k_{[-1]}$ LSBs. From (7) it can be seen that $\widehat{M}_{[-1]}$ contains a smaller percentage of zeros than either $M_{[-1]}$ or $M_{[-2]}$; as a consequence, it is slightly less compressible. This is the price to be paid for error resilience.

3.2. Decoding

To exploit the side information and to recover the MSBs of $x_{m,n,i}$, the decoder needs to know a model to predict the values of the current band by the side information, such that the prediction error is not larger than the maximum prediction error estimated by the encoder. A linear predictor is used by the decoder, but the estimation of the average of the current block and the quantized Least-Squares estimation value of eq.(2) is required. The μ_i average is estimated by resorting to the pixels of spatially contiguous blocks already decoded and to those belonging to the side information. Let us denote as $\mu_{i(u)}$ and $\mu_{i(l)}$ the average values of the blocks above and on the left of the current block, and $\mu_{i-1(u)}$ and $\mu_{i-1(l)}$ the corresponding values of the side information. The average value can be estimated by the following equation

$$\widehat{\mu}_i = \frac{1}{2}\Big[\mu_{i(u)}\Big(1+\frac{\mu_{i-1}-\mu_{i-1(u)}}{\mu_{i-1(u)}}\Big)+\mu_{i(l)}\Big(1+\frac{\mu_{i-1}-\mu_{i-1(l)}}{\mu_{i-1(l)}}\Big)\Big]. \tag{8}$$

A large number of predictors is built by perturbing the estimated values of μ_i, and for each $\widehat{\mu}_i$ a set of 200 uniform levels from 0 to 1.99 is defined to estimate $\widehat{\alpha}_{[-1]}$. All the candidate predicted values obtained by applying all predictors to the side information are combined with the LSBs. The reconstruction is validated checking the following two conditions:

- for all pixels of the block, $|x_{m,n,i} - \widetilde{x}_{m,n,i}| < 2^{k_{[-1]}-1}$
- the CRC obtained from the reconstructed pixels is identical to the received CRC.

If the previous band $x_{m,n,i-1}$ is not available as side information, e.g. because it has been corrupted by errors, the decoding process is performed using $x_{m,n,i-2}$; the process will be successful if $k_{[-2]} \le k_{[-1]}$. On average, we have seen that the decoding process based on $x_{m,n,i-2}$ is successful about 70% of the times.

4. EXPERIMENTAL RESULTS

The compression performance of the proposed scheme has been compared with that of JPEG-LS,[22] 2D-CALIC[23] and 3D-CALIC.[5] The results on AVIRIS scenes are reported in Tab. 1. As can be seen, 2D-CALIC performs better than JPEG-LS, and 3D-CALIC obtains the best performance overall. The proposed scheme outperforms JPEG-LS by about 1.7 bpp, and 2D-CALIC by more than 1.5 bpp. However, the performance gap with respect to 3D-CALIC is about 0.5 bpp, which is the price to be paid for using a simple scalar M-ary S-W coder.

Table 1. Coding efficiency (bit-rate) of the proposed algorithm on AVIRIS scenes, compared to that of conventional 2D and 3D schemes.

algorithm	*Cuprite*	*Moffett*	*Jasper*	*Average*
JPEG-LS	6.90	7.65	7.61	7.39
2D-CALIC	6.76	7.56	7.49	7.27
3D-CALIC	5.13	5.23	5.14	5.17
Proposed	5.59	5.83	5.71	5.71

We also tested the proposed algorithm on five raw (as opposed to calibrated) scenes. The first four scenes (Sc0, Sc3, Sc10 and Sc11) are on 16 bpp, while the "Hawaii" scene is on 12 bpp. The results are shown in Tab. 2. As can be seen, the bit-rates achieved on the 16 bpp raw scenes are larger than on the calibrated scenes, as the raw scenes contain more information. On the other hand, the bit-rate achieved on the "hawaii" scene is significantly smaller, as this raw scene is represented on only 12 bits.

The encoder complexity, measured as sunning time on a Linux workstation, is about 1.85 times larger than JPEG-LS, and much lower than 3D-CALIC.

As for error resilience, as has been said, the proposed algorithm is able to reconstruct the current band exactly, using the second previous band, about 70% of the times, prividing a high degree of robustness.

5. CONCLUSIONS

We have proposed a compression algorithm for hypersectral images featuring good coding efficiency, low encoder complexity, and error resilience. This algorithm performs significantly better than JPEG-LS and 2D-CALIC, with much smaller complexity. Therefore, it is a viable option for on-board real-time image compression.

Table 2. Coding efficiency (bit-rate) of the proposed algorithm on raw AVIRIS scenes.

Scene	Bit-rate
Sc0	7.03
Sc3	6.88
Sc10	6.18
Sc11	6.62
Hawaii	3.62

ACKNOWLEDGMENTS

This work has been sponsored by ESA-ESTEC under ITI type-B project 21289/07/NL/CB "PILL - Parallel implementation of lossless hyperspectral image compression".

Part of the activities presented in this paper have been carried out using the JPEGLS-E IP core by Alma Technologies Inc.

REFERENCES

1. B. Huang and Y. Sriraja, "Lossless compression of hyperspectral imagery via lookup tables with predictor selection," in *Proc. SPIE, vol. 6365*, 2006.
2. J. Mielikainen, "Lossless compression of hyperspectral images using lookup tables," *IEEE Signal Processing Letters*, vol. 13, no. 3, pp. 157–160, Mar. 2006.
3. B. Penna, T. Tillo, E. magli, and G. Olmo, "Transform coding techniques for lossy hyperspectral data compression," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1408–1421, May 2007.
4. K. Anderson, "Low-cost, radiation-tolerant, on-board processing solution," in *Proc. of IEEE AerospaceConference*, 2005.
5. X. Wu and N. Memon, "Context-based lossless interband compression - extending CALIC," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 994–1001, June 2000.
6. Z. Xiong, A.D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–94, Sept. 2004.
7. E. Magli, M. Barni, A. Abrardo, and M. Grangetto, "Distributed source coding techniques for lossless compression of hyperspectral images," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, 2007.
8. E. Baccaglini, M. Barni, L. Capobianco, A. Garzelli, E. Magli, F. Nencini, and R. Vitulli, "Low-complexity lossless compression of hyperspectral images using scalar coset codes," in *Proceedings of Picture Coding Symposium*, 2007.
9. D. Slepian and J.K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, July 1973.
10. A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.
11. S.S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1181–1203, May 2003.
12. R. Zamir, "The rate loss in the Wyner-Ziv problem," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 2073–2084, Nov. 1996.
13. R. Puri and K. Ramchandran, "PRISM: a "reversed" multimedia coding paradigm," in *Proc. of IEEE International Conference on Image Processing*, 2003, pp. 617–620.
14. Q. Xu and Z. Xiong, "Layered Wyner-Ziv video coding," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3791–3803, Dec. 2006.
15. H. Wang and A. Ortega, "Scalable predictive coding by nested quantization with layered side information," in *Proceedings of IEEE International Conference on Image Processing*, 2004, pp. 1755–1758.
16. N.-M. Cheung, C. Tang, A. Ortega, and C.S. Raghavendra, "Efficient wavelet-based predictive Slepian-Wolf coding for hyperspectral imagery," *Signal Processing*, vol. 86, no. 11, pp. 3180–3195, Nov. 2006.
17. S.S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626–643, Mar. 2003.

18. J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Communications Letters*, vol. 5, no. 10, pp. 417–419, Oct. 2001.
19. A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, Oct. 2002.
20. M. Grangetto, E. Magli, and G. Olmo, "Distributed arithmetic coding," *IEEE Communications Letters*, vol. 11, no. 11, pp. 883–885, Nov. 2007.
21. D.C. Feldmeier, "Fast software implementation of error detection codes," *IEEE/ACM Transactions on Networking*, vol. 3, no. 6, pp. 640–651, 1995.
22. M.J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.
23. X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Transactions on Communications*, vol. 45, no. 4, pp. 437–444, Apr. 1997.