Band Ordering in Lossless Compression of Multispectral Images

Stephen R. Tate*

Department of Computer Sciences
University of North Texas
P.O. Box 13886
Denton, TX 76203–6886

Abstract

This paper examines the compression benefits that can be obtained by reordering the bands of a multispectral image. In particular, we consider a model of lossless image compression in which each band of a multispectral image is coded using a prediction function involving values from a previously coded band of the compression. Clearly, the ordering of the bands determines which bands can be used for prediction, and this, in turn, can strongly influence compression performance.

We present an efficient algorithm for computing the optimal band ordering for a multispectral image. This algorithm has time complexity $O(n^2)$ for an n-band image, while the naive algorithm takes time $\Omega(n!)$. We also define a slight variant of the optimal ordering problem that is motivated by some practical concerns on band extraction, and prove that this problem is NP-hard, and hence computationally infeasible, in all cases except for the most trivial possibility.

In addition, we report on our experimental findings using the algorithms designed in this paper applied to real multispectral satellite data. The results show that for certain image sources, such as the CZCS and AVIRIS satellite sensors, choosing the optimal band order as described in this paper can produce substantial improvements in the compression ratio. In addition, the experiments show that for CZCS data (and presumably data from other sources) a single band ordering can be selected that provides almost optimal compression performance for any data from that source, eliminating the need to compute the optimal ordering for each individual data set.

*This research was supported by NASA subcontract 550-63 of prime contract NAS5-30428, and a substantial part of the experiments were conducted when the author was affiliated with Duke University and visiting the CESDIS office at the Goddard Space Flight Center.

Sensor Names	
AVHRR	Advanced Very High Resolution Radiometer 10 bits per pixel, 5 bands
AVIRIS	Airborne Visible/Infrared Imaging Spectrometer 12 bits per pixel, 210 bands
CZCS	Coastal Zone Color Scanner 8 bits per pixel, 5 or 6 bands
TM	Thematic Mapper 8 bits per pixel, 7 bands

Table 1: Information on various multispectral image sources.

1 Introduction

Multispectral satellite images require enormous amounts of space, and with NASA's project EOS (the Earth Observing System) data will be generated at an unprecedented rate. The estimates are that over a terabyte (10^{12} bytes) of data will be generated every day by the EOS satellites, most of it multispectral image data. Largely due to this fact, a lot of attention has recently been focussed on compression of multispectral images [1, 2, 3, 4, 5]. However, most of the compression methods that exploit spectral as well as spatial redundancy have been lossy compression algorithms, and for archival storage and for certain applications it is important to use lossless compression in order to preserve all of the data that is collected. One notable exception to this is the work of Roger and Cavenor [4] who extensively study various prediction and coding methods used for lossless compression of AVIRIS data. Table 1 lists some current, widely used multispectral sources, with their acronyms, full names, and basic properties — it is data from these sensors that we used in this study.

In this paper, we study lossless compression of multispectral images. Spectral redundancy is extracted by coding each band of the multispectral image by making use of a second "prediction band". In much the same way that standard single-image lossless compression is separated into the two separate components of prediction and coding, we divide the lossless compression of multispectral images into three components: band ordering, prediction, and coding. The new stage, band ordering, refers to selecting a permutation of the bands in which the bands that are coded first act as good predictors for the later bands in the ordering. The band ordering phase is independent of the other phases of the compressor, and the computational problems associated with this phase are identified and studied in this paper. In particular, given any particular predictor and coder (such as those in this paper or those studied by Roger and Cavenor [4]), we give an efficient algorithm for computing the optimal ordering of the bands for compression performance. For an n-band multispectral image, our algorithm runs in $O(n^2)$ time, which is a vast improvement over the obvious $\Omega(n!)$ solution. In timing tests performed on a DEC Alpha workstation, our algorithm computed the optimal ordering from the input matrices (see Section 2) of a 12 band image in 1/100 of a second, whereas the naive algorithm took 87 minutes; more dramatically, our algorithm took approximately 7 seconds to find the optimal ordering for all 210 bands of the AVIRIS data set, whereas the naive algorithm would take considerably more time than the universe has been in existence!

In some cases, however, the optimal ordering may not be a practical archival format for large numbers of bands (see Section 3 for more information on this), so we consider the natural restriction on band orderings that overcomes this problem. We refer to the restricted ordering as the *optimal compression order with bounded prediction*. Unfortunately, this new problem is shown to be NP-hard except in the most trivial case, so is computationally infeasible.

Next, we examine various possibilities for the prediction phase for multispectral data. Most predictors for image data use some form of linear prediction, and most of these use fixed coefficients for the prediction (or take coefficients from a small set of possibilities, as in lossless JPEG compression [6]). With multispectral images it is vital that the linear prediction coefficients be chosen adaptively, and different methods for doing this are discussed in Section 4.

The coder for multispectral images is the simplest part, and is discussed in Section 5. We investigate the use of various configurations of an arithmetic coder, using data within both the current band and the previous "prediction band" as context for the coder.

Finally, in Section 6 we present some experimental results using the techniques and algorithms presented in this paper on real multispectral data obtained from NASA. The experiments show that the compression methods we developed can have a substantial practical impact on compression performance. In addition, our experiments show that using a fixed, precomputed, band ordering can achieve almost optimal performance on a large class of data; therefore, using a pre-computed band ordering can give a very fast, high-performance compression system for multispectral images.

2 Terminology and Notation

Just as single-image data can be represented by a two-dimensional array $S[x, y]$, the data of a multispectral image can be represented as a three-dimensional array, with entry $M[b, x, y]$ representing the pixel from band b, row x, column y. In traditional single-image compression, when coding pixel $S[x, y]$, the previously coded neighboring pixels may be used to predict the value of the current pixel. Likewise, when coding pixel $M[b, x, y]$ for a multispectral image, the previously

$$B = \begin{pmatrix} 122078 & 95331 & 111052 & 138922 \\ 121696 & 95785 & 111046 & 138897 \\ 121834 & 95386 & 111111 & 138899 \\ 122046 & 95754 & 110982 & 138943 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 79321 & 93959 & 137762 \\ 104809 & 0 & 87687 & 135151 \\ 104836 & 73263 & 0 & 133614 \\ 121742 & 93657 & 107673 & 0 \end{pmatrix}$$

Figure 1: Sample A and B matrices (from Thematic Mapper data, bands 1–4).

---

coded neighboring pixels may be used for prediction, but we also allow neighboring pixels to come from a previously coded band. In other words, for any band b, we choose a *prediction band* $p(b)$ and use pixel values from band $p(b)$ in the encoding of $M[b, x, y]$.

The compression methods described in this paper all code the pixels of a band in scan order. At any point in time, the pixel $M[b, x, y]$ that the compression algorithm is predicting or coding is called the "current pixel", which comes from the "current band" (band b). As mentioned above, we also designate a separate band $p(b)$ as the "prediction band" for the current band. Just as in single-image compression, all pixel values used in prediction must be coded before the current pixel, including pixel values from the prediction band. Thus, we may use the prediction band's pixel location corresponding to the current pixel (i.e., $M[p(b), x, y]$) if and only if band $p(b)$ is coded before band b. Clearly, the ordering of the bands determines what bands may be selected as the prediction band for the current band, and determining an ordering of the bands that maximizes compression over all possible orderings is a fundamental problem.

For any given predictor/coder pair, and two bands b_1 and b_2, we can define the following two values:

B_{b_1,b_2} is the compressed size of band b_2 if band b_1 is used for prediction, but $M[b_1, x, y]$ is *not* used in the prediction of $M[b_2, x, y]$. In other words, this is the compressed size for band b_2 using b_1 for prediction, even if band b_2 is encoded *before* band b_1.

A_{b_1,b_2} is the compressed size of band b_2 if band b_1 is used for prediction, and the current pixel location from band b_1 is used in the prediction. This is the compressed size possible for band b_2 if band b_2 is encoded *after* band b_1.

We will represent the A and B values for a multispectral image as matrices. See Figure 1 for example A and B matrices, which were obtained from bands 1–4 of the Thematic Mapper data of the Washington D.C. area.

In the next section, we will examine various ways of choosing a band ordering when given matrices A and B.

3 Determining Band Compression Order

In this section, we examine the following problem: In what order should we compress the bands of a multispectral image to achieve the best compression? For example, if band 2 acts as a good predictor for band 1, substantial savings may be achieved by compressing band 2 first. We will show that, given the matrices as defined in the previous section, the optimal ordering of the bands can be found very efficiently. For multispectral data with a small number of bands, this result shows that optimal ordering is both simple and practical.

However, for data with a large number of spectral bands (such as the 210 band AVIRIS data), the optimal ordering can be computed, but may not be what is desired. In particular, given a compressed file containing all 210 bands of AVIRIS data, it may be desirable to be able to extract one particular band from the compressed data. Unfortunately, the band dependencies of the optimal ordering may be a single, long chain, which would mean that *all* bands would have to be uncompressed in order to uncompress the last single band in the chain. A good solution to this problem would be to partition the bands into small sets, each containing at most b bands, and then when compressing one band, the only other bands that may be used for prediction are those within the same set. With this method, at most b bands would have to be uncompressed in order to extract a single band from the compressed file. For a particular partition size bound b, we will call the optimal solution to this problem the *optimal compression order with bounded prediction.*

We will show that finding the optimal compression order with bounded prediction is easy if $b = 2$, but is NP-hard, and hence computationally intractable, for any $b \geq 3$.

3.1 Unconstrained Optimal Ordering

In this section we consider the problem of finding an optimal compression order, with no constraints on how many bands are necessary for uncompression of a single band. This ordering is the best possible ordering for encoding bands, and is practical when either there is a small number of bands (such as the 5 bands of CZCS data or the 7 bands of TM data), or if extraction of a single band is never needed (an archive of AVIRIS data may require uncompression of all 210 bands when it is accessed).

We will transform the problem of finding an optimal compression ordering into a problem on weighted directed graphs which has a known efficient solution. The directed graph $G = (V, E)$ is constructed so that each edge $E_{i,j}$ has a corresponding weight $W_{i,j}$ that represents the compression savings attainable by compressing band i before band j. To construct this graph, define $B_{\min,j}$ as

$$B_{\min,j} = \min_{1 \le i \le n} B_{i,j}.$$

$B_{\min,j}$ represents the minimum compressed size of band j if the current pixel location is not used from any other band. In particular, a compressed size of $B_{\min,j}$ is attainable for band j regardless of the order in which the bands are compressed.

Of course, if the bands are ordered such that band i is encoded before band j, then the current pixel location from band i can be used to predict the current pixel of band j. In some cases, the resulting compressed size may be substantially smaller than the size that was attainable without considering ordering. We define the edge weights of our graph in terms of the savings possible by encoding band j using the current pixel location from band i for prediction. In particular,

$$W_{i,j} = \begin{cases} 0 & \text{if } A_{i,j} \ge B_{\min,j}, \\ B_{\min,j} - A_{i,j} & \text{otherwise.} \end{cases}$$

The problem of finding an optimal compression order is equivalent to the problem of finding a maximum weight directed spanning forest of G. The resulting spanning forest defines a partial order on the vertices. Since the spanning forest is maximum weight, the partial order represents the maximum compression savings of any band ordering, or the optimal compression order of the bands of the multispectral image. It is known that a maximum weight directed spanning forest (also known as an "optimal branching") can be found in $O(|V| \log |E| + |E|)$ time on sparse graphs and $O(|V|^2)$ time on dense graphs [7, 8], so it follows that an optimal compression order can be found in $O(n^2)$ time.

The graph G that is derived from the sample A and B matrices of Figure 1 is shown in Figure 2, where dashed and solid lines both represent graph edges, and the solid lines are the edges that are part of the maximum weight directed spanning forest. Thus, the solid lines describe the optimal compression ordering for these four bands. Band 2 is coded first, and then band 1 is coded using band 2 for prediction; band 3 is coded using band 2 for prediction; and band 4 is coded using band 3 for prediction (of course, this is just one arbitrarily chosen ordering that is consistent with the partial order defined by the spanning tree). The total weight of the selected spanning forest is 45,257, meaning that this band ordering can save 45,257 bytes over an ordering-independent coding scheme. We summarize the result of this section in the following theorem.

Theorem 3.1 Given values $A_{i,j}$ and $B_{i,j}$ for all pairs of bands (i, j) from an n-band multispectral image, the optimal compression ordering can be computed in $O(n^2)$ time.

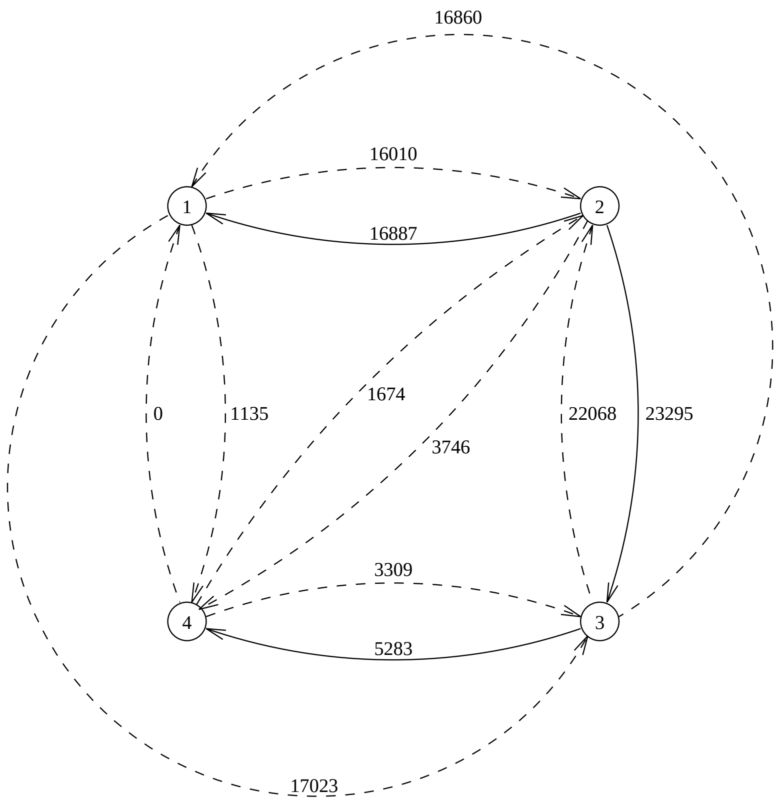


Figure 2: Graph derived for sample A and B matrices; spanning tree edges are the solid lines.

3.2 Optimal Ordering with Bounded Prediction

As mentioned at the beginning of this section, the optimal ordering is not always practical. In particular, if it is necessary to be able to extract and uncompress a single band of the image efficiently, the optimal compression ordering does not always make this possible. Consider a case where the directed spanning tree, obtained as in the previous subsection, is actually a long directed path. In a 210 band data set, it would be possible for band 210 to be coded using band 209 for prediction, band 209 to be coded using band 208, etc. Thus, in order to extract band 210, all of the 209 other bands would have to be uncompressed in addition to band 210.

For data sets with a large number of bands, a reasonable approach would be to partition the bands into small sets (containing, say, b bands each), and then compress each set using its optimal compression ordering. In this case, if a single band needs to be extracted, only the other bands in the same set as the requested band will possibly need to be uncompressed. We call such an ordering the *optimal compression order with bounded prediction*, and prove the following theorem regarding the complexity of computing such an ordering.

Theorem 3.2 Given values $A_{i,j}$ and $B_{i,j}$ for all pairs of bands (i,j) from an n-band multispectral image, the problem of finding an optimal compression order with bounded prediction is NP-hard if $b \geq 3$, but can be computed in $O(n^3)$ time if $b = 2$.

Proof: To show that the problem of finding an optimal compression order with bounded prediction is NP-hard for $b \geq 3$, we will show that the following decision problem is NP-complete.

COMPRESSION ORDER WITH BOUNDED PREDICTION (COWBP): Input to this problem are the matrices A and B, and a compressed size bound C. The problem is to decide whether there exists a partitioned compression ordering for the bands such that no partition contains more than 3 bands, and the total compressed size is at most C.

This problem is clearly in NP, and to show that it is NP-complete, we will reduce the PARTITION INTO PATHS OF LENGTH TWO (PIPLT) to COWBP. PIPLT, which is known to be NP-complete [9], takes an undirected graph with $n = 3q$ vertices as input, and asks whether or not there exists a partition of the graph into paths of length two.

Given a graph $G = (V, E)$ that is input to the PIPLT problem, we convert the graph into an instance of COWBP by setting the inputs as follows:

$$\begin{aligned} B_{i,j} &= 2 \text{ for all } 1 \leq i,j \leq n \\ A_{i,j} &= \begin{cases} 1 & \text{if } (i,j) \in E, \\ 2 & \text{if } (i,j) \notin E, \end{cases} \\ C &= \tfrac{4}{3}n \end{aligned}$$

Next, we show that there is a partitioned compression ordering satisfying the conditions of this instance of COWBP if and only if the original graph G can be partitioned into paths of length two.

First, assume that G can be partitioned into paths of length two, where the paths are given by triples $< v_{i,1}, v_{i,2}, v_{i,3} >$ for $i = 1, 2, \cdots, q$. These triples define the partition of the COWBP input, and the "bands" can be compressed by coding $v_{i,2}$ using $v_{i,1}$ (without the current pixel), $v_{i,1}$ using $v_{i,2}$ (with the current pixel), and then $v_{i,3}$ using $v_{i,2}$ (with the current pixel). The resulting compressed size for this set i of the partition is therefore $B_{v_{i,1},v_{i,2}} + A_{v_{i,2},v_{i,1}} + A_{v_{i,2},v_{i,3}} = 2+1+1 = 4$. Since there are $q = \frac{n}{3}$ sets in the partition, the total compressed size for all bands is exactly $\frac{4}{3}n$, so if the original problem is in PIPLT, then the computed problem is in COWBP.

Next, assume that the computed problem is in COWBP, so there is a partitioned compression ordering with total compressed size at most $\frac{4}{3}n$. We can choose an optimal partitioned compression ordering such that band i is compressed using the current pixel of band j only if $A_{i,j} = 1$ (otherwise, the current pixel is not required since $B_{i,j} = A_{i,j} = 2$). Furthermore, we can assume that any two bands in the same partition are in the same connected component of the ordering, since if this were not the case we could split the partition into smaller independent partitions that did have the desired property. Let c_i be the number of sets in the partition with size i (for $1 \leq i \leq 3$). Under our assumptions, the compressed size of any set of size 1 is 2, any set of size 2 is 3, and any set of size 3 is 4, so the total compressed size is

$$2c_1 + 3c_2 + 4c_3 = (c_1 + 2c_2 + 3c_3) + (c_1 + c_2 + c_3) = n + (c_1 + c_2 + c_3).$$

The only way to have $c_1 + c_2 + c_3 \leq \frac{n}{3}$ is for $c_1 = c_2 = 0$ and $c_3 = \frac{n}{3}$, so all partitions must have exactly 3 bands, which must correspond to paths of length two in the original graph. Therefore, if the computed problem is in COWBP, then the original problem is in PIPLT, which completes the NP-completeness proof.

A simple padding argument shows that COWBP remains NP-complete when generalized to any constant $b > 3$.

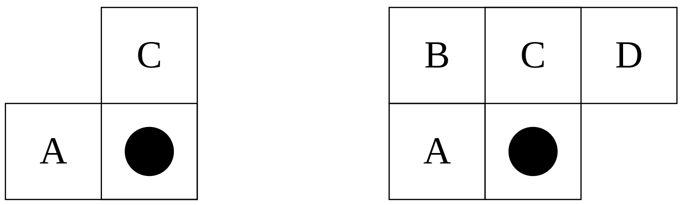
When we allow the partition to contain only sets of size two or smaller (i.e., $b = 2$), the encoding size of bands i and j in the same set would result in a compressed size of exactly

$$C_{i,j} = \min(B_{i,j} + A_{j,i}, A_{i,j} + B_{j,i}).$$

Let $M = \max(B_{i,j})$, and create a complete undirected graph with weights $W_{i,j} = 2M - C_{i,j}$. Finding a maximum weight matching on this graph gives the optimal partitioning of the bands into sets of size 2, which also gives us exactly the optimal compression ordering with bounded prediction ($b = 2$). It has been known since 1976 that such a matching can be found in $O(n^3)$ time [10]. It should be noted that algorithms for matching are known that take advantage of input instances with bounded integral weights on the edges, and beat the $O(n^3)$ time bound with slightly better asymptotic results [11]. Without explicit bounds on the compressed size of the bands of our multispectral image, it is impossible to state our results in terms of the improved algorithms, and we refer the interested reader to the algorithm reference [11]. ▮

4 Prediction for Multispectral Images

The vast majority of lossless image compression algorithms are made up of a prediction stage in which pixel values are predicted, and a coding stage in which the difference between the actual pixel value and the predicted value is coded (for recent examples, see [12, 6]). The predictor typically uses a linear function to predict the current pixel value, based on the values of previously coded pixels (the "neighborhood" of the current pixel). In compression algorithms that code the pixels



(a) 2-neighborhood (b) 4-neighborhood

Figure 3: Neighboring pixels used in prediction phase.

in scan-order, the neighborhood is usually a subset of the 4-neighborhood shown in Figure 3(b), where the current pixel is marked with the solid circle. For example, we may predict the current pixel by computing the average of its two nearest neighbors: $(A+C)/2$. The lossless JPEG scheme has 8 different possible linear prediction functions of A, B, and C, and is free to use any of those eight [6].

Since the relationship between different bands of a multispectral image is not known ahead of time, we must find a good prediction function by examining the image data. In particular, in our experiments we consider both 2-neighborhoods and 4-neighborhoods for pairs of bands (see Figure 3), and use the actual image data to determine the coefficients of the linear function of neighborhood values that minimizes the variance of the prediction error. Determining the coefficients in this way is similar (although not exactly the same) to standard linear prediction methods (see for example [13]), and can be done using least squares approximation algorithms.

The experiments of this paper were done with a predictor that finds linear coefficients for the 2-neighborhood and 4-neighborhood of the current pixel location, where values from both the current band and the prediction band are used in the prediction function. Both the 2-neighborhood and 4-neighborhood are then used for a second pass on the data using the computed coefficients, and whichever resulted in a smaller compressed size is reported. The program computes the coefficients both with and without the current pixel location in the prediction band, and the resulting compressed sizes are used for the A and B matrix values, respectively.

5 Coding for Multispectral Images

The coder has the responsibility of turning the predicted values into an output bit-stream. Various approaches have been taken in different compression systems, including Huffman coding, Rice coding, and arithmetic coding (or variants such as Q-coding). A good explanation of coding in general, including Huffman coding and arithmetic coding, can be found in [14]; Rice coding is described in [15], which is similar to Golumb coding as described in [16]. Huffman coding performs poorly on small alphabets or when the prediction phase produces very small (or low entropy) errors, and Rice coding performs poorly for small errors or when the errors are not generated according to a Laplacian distribution. While the distribution of errors is usually close to Laplacian, it was discovered in our experiments that CZCS data produced very markedly non-Laplacian error distributions (the other sensors produced prediction errors that were more or less Laplacian).

Our final choice for a coder was an adaptive arithmetic coder with three neighboring prediction errors used as context for the coder (the two nearest neighbors in the current band, and the current pixel location from the prediction band when possible). While the actual prediction errors could easily be used for neighboring pixels in the same band, the prediction error for the current location in the prediction band was estimated by using the simple single-image linear prediction formula, $X - \lfloor (A+C)/2 \rfloor$, where X is the prediction band pixel in the current location, and A and C are neighboring pixels in the prediction band as defined by Figure 3(a). The prediction errors were bucketed (similar to [17]) into "zero", "small" (1-2), "medium" (3-8), and "large" (over 9), with the sign of the error also used in the context. Since the context is computed from 3 prediction errors with 7 possible values each, the coder uses 343 contexts.

The actual prediction errors are encoded by first encoding a tag denoting negative, positive, or zero error, and then encoding the error magnitude two bits at a time. This method substantially reduces the amount of storage required for maintaining the error frequency statistics, since frequency tables only need to be maintained for a 4 symbol alphabet (the 2-bit blocks). For example, in an image with 16 bit pixels, the error can range from -65,535 to +65,535, so directly coding the prediction error would require 131,071 frequency counts per context (for a total of almost 4.5 million counts over all contexts). On the other hand, keeping statistics only for the error sign and the two-bit blocks of the error magnitude requires only 35 frequency counts per context (for a total of 12,005 frequency counts over all contexts). It is very significant to note that when the errors are Laplacian distributed (the majority of cases), encoding the errors in this significantly more efficient way is fully as efficient as keeping separate frequency counts for all possible error values. Furthermore, when the errors are not Laplacian distributed (such as for the CZCS data), this method greatly outperforms true Laplacian coders such as the Rice coder, and performs almost

Data Set	Optimal Order	Paired Code	Independent	JPEG	"Compress"
czcs 486x1536x5	2.95:1 2.71	2.61:1 3.06	2.37:1 3.37	2.46:1 3.25	2.02:1 3.97
neworl 512x512x7	2.68:1 2.98	2.58:1 3.10	2.47:1 3.24	2.49:1 3.22	1.87:1 4.29
ridgely 468x368x7	2.78:1 2.87	2.59:1 3.09	2.50:1 3.20	2.49:1 3.21	1.91:1 4.19
washdc 512x512x7	2.58:1 3.10	2.45:1 3.26	2.32:1 3.44	2.30:1 3.48	1.86:1 4.31
aviris 512x614x210	3.53:1 4.53	3.21:1 4.98	2.98:1 5.37	2.90:1 5.53	2.07:1 7.73
avhrr1 3962x409x5	2.77:1 5.78	2.76:1 5.80	2.77:1 5.78	2.50:1 6.40	1.53:1 10.45
avhrr2 3502x409x5	2.61:1 6.13	2.61:1 6.14	2.61:1 6.13	2.40:1 6.67	1.47:1 10.87

Table 2: Results from first experiment: Top line is compression ratio, bottom line is bits per pixel.

as well as the arithmetic coder with full frequency counts.

6 Experimental Results

We performed many experiments using the algorithms described in this paper and real multispectral satellite data. Our experiments show that band reordering can substantially improve compression performance for certain data sets, and that using a fixed, pre-computed band ordering works well for entire classes of data, giving a very fast and high-quality compression system for multispectral images. The data for our experiments came from the four types of sensors listed in Table 1.

The first set of images, representing a cross-section of the image sources, contains TM data "neworl", "ridgely", and "washdc", 5 bands of CZCS data "czcs", AVHRR data "avhrr1" and "avhrr2", and an AVIRIS data set denoted "aviris". Most of these data sets are available by anonymous ftp from site `chrpalg.gsfc.nasa.gov`, so are available to other researchers for comparison purposes.

To compute the A and B matrices, the entries were computed in parallel on a CM-5 parallel computer. Once these values were computed, the A and B matrices were fed into implementations of the algorithms described in Section 3 for finding the optimal band ordering, and for finding the optimal paired ordering. The results are given in terms of compression ratio and encoded bits per pixel in Table 2, and the compression ratios are shown graphically in Figure 4. The compression ratio using our prediction and coding phases, but with no inter-band prediction, is

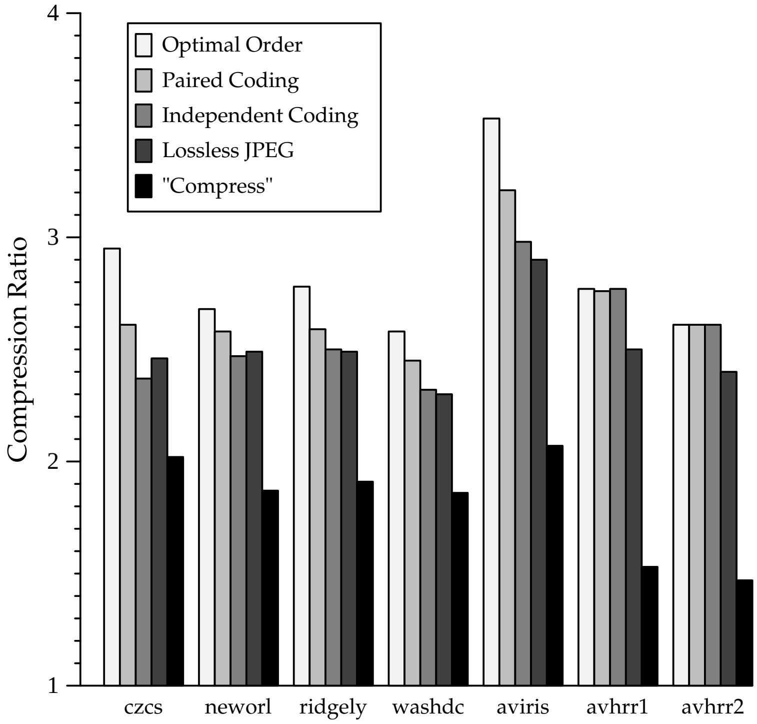


Figure 4: Results (compression ratio) from first experiment.

shown as "Independent Coding" (meaning that the bands are coded independently of other bands). In addition, the compression ratios found using lossless JPEG (with each band coded independently of the others) and the UNIX compress command are included for comparison purposes. The performance of lossless JPEG compares well with the best lossless single-image compressors. The UNIX compress results are included since many installations currently choose this method for compressing their multispectral data sets, and these results emphasize how poor this choice is.

As can be seen from Table 2 and Figure 4, the effect of band ordering depends heavily on the source of the data. In particular, large gains were seen by selecting the optimal band ordering for the AVIRIS and CZCS data, moderate gains were seen in the TM data sets, and almost nothing was gained by band-ordering in the AVHRR data. This suggests that the individual bands of the AVHRR data are more independent than the bands of the other sensors, which is supported by the technical specifications of the sensor.

It is interesting to note that for data sets in which band ordering made a difference, the performance of paired ordering is only a moderate improvement over independent coding. This implies that the band dependencies are of a more global nature than pairs of bands. In light of our proof that computing the optimal partitioning into blocks of size three or more is NP-hard, it seems

Blocking Method	Compression Ratio
No blocking (optimal order)	3.53:1
Blocks of 5 adjacent bands	3.39:1
Optimal partitioning of 2-band blocks	3.21:1
Block size 1 (independent coding)	2.98:1

Table 3: Partitioning band by position, without regard to content.

that close-to-optimal performance is not possible without using the absolute optimal ordering and risking long dependence chains. To examine the importance of selecting a good partitioning, we divided the 210 bands of AVIRIS data into blocks of 5 adjacent bands per block, without regard for the content of those bands. In other words, the blocks consisted of bands 1-5, bands 6-10, bands 11-15, etc. Within each block, the bands were optimally ordered, but there is no reason to believe that the chosen blocks bear any resemblance to the optimal partitioning. The results are shown in Table 3.

It can be seen from Table 3 that simply picking blocks of adjacent bands for the partitioning seems to work well for AVIRIS data. While the compression ratio does not quite match the performance of the optimal ordering, the added benefit of allowing efficient extraction of a single band from the compressed archive may be worth sacrificing some compression.

The next experiment we conducted was to determine how much the band ordering depends on the sensor, and how much it depends on the individual data sets. For this test, we obtained seven independent data sets from a single sensor (CZCS) and ran all of the previous tests, in addition to the following test: for any particular data set, the optimal compression ordering for the *other* six data sets is computed, and then that ordering is used for the data set being tested. The results, with the last type of test labeled as "Pre-Computed Order", are shown in Table 4 and Figure 5.

As can be seen from these results, in most cases the order computed from the other images gave a compression ratio that was competitive with the optimal order computed specifically for that image. This suggests that, at least for CZCS data, the band ordering is more a function of the sensor than of the individual image; therefore, a reasonable approach to take in a production compression package would be to compute the optimal ordering from a large, representative set of images, and hard-code this band ordering into the compressor. To compress a multispectral image with this approach, no band-ordering has to be computed, and the A and B matrices do not need to be computed, so the compression is very efficient. The results of this last experiment show that this approach should yield almost optimal compression.

The optimal ordering for CZCS data, computed from our seven data sets, and the optimal

Data Set	Optimal Ord	Pre-Comp Ord	Paired Code	Ind.	JPEG	“Compress”
czcs1 970x1968x6	3.34:1 2.40	3.27:1 2.44	2.92:1 2.74	2.71:1 2.96	2.92:1 2.74	2.12:1 3.77
czcs2 962x1968x6	2.39:1 3.35	2.37:1 3.37	2.18:1 3.67	2.02:1 3.96	2.15:1 3.71	1.68:1 4.77
czcs3 970x1968x6	3.28:1 2.44	3.28:1 2.44	3.00:1 2.66	2.80:1 2.85	3.02:1 2.65	2.39:1 3.35
czcs4 970x1968x6	3.43:1 2.34	3.43:1 2.34	3.11:1 2.57	2.82:1 2.84	2.99:1 2.67	2.41:1 3.32
czcs5 850x1968x6	2.76:1 2.90	2.67:1 2.99	2.53:1 3.16	2.36:1 3.38	2.40:1 3.34	1.71:1 4.69
czcs6 970x1968x6	2.75:1 2.91	2.75:1 2.91	2.52:1 3.17	2.26:1 3.54	2.35:1 3.41	1.87:1 4.28
czcs7 970x1968x6	2.51:1 3.18	2.51:1 3.18	2.27:1 3.52	2.06:1 3.89	2.12:1 3.77	1.70:1 4.71

Table 4: Results from CZCS experiments: Top line is compression ratio, bottom line is bits per pixel.

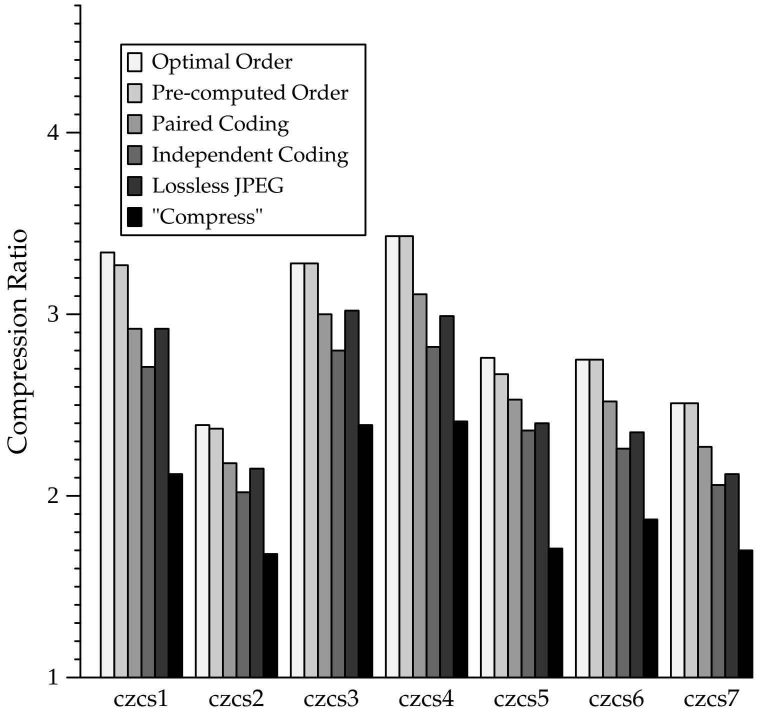


Figure 5: Results (compression ratio) from CZCS experiments.

CZCS Data	
Band 6	Alone
Band 1	Using band 6
Band 2	Using band 1
Band 3	Using band 2
Band 4	Using band 3
Band 5	Using band 1

TM Data	
Band 2	Using band 1 without current location
Band 1	Using band 2
Band 3	Using band 2
Band 7	Using band 3
Band 5	Using band 7
Band 6	Using band 7
Band 4	Using band 5

Table 5: Globally optimal band orderings for TM and CZCS sensors.

ordering for the TM data, computed from data sets "washdc" and "neworl", are shown in Table 5.

7 Conclusions

In this paper, we have extensively studied the benefits and problems associated with reordering the bands of a multispectral image for performing lossless compression. Under a reasonable model of inter-band prediction, we have shown how to efficiently compute the optimal compression band ordering. In addition, we have formalized the restrictions that arise when bands need to be extracted individually from a compressed archive, and have shown that computing the optimal ordering under these restrictions is NP-hard, except in the most simple case.

Our experiments show that for certain data sources (such as CZCS or AVIRIS imagery), the band ordering algorithms of this paper can substantially improve compression performance. Interestingly, some data sources (such as the AVHRR data) do not benefit from the techniques of this paper at all, implying that the bands are independent, and can be treated separately (as if the data was not a multispectral image at all). By computing the band ordering as described in this paper and evaluating the compression results, it is easy to determine if band re-ordering is going to have a substantial benefit for a particular data set.

While computing the optimal band ordering may be efficient (i.e., polynomial time), it is certainly not fast, requiring that all pairs of bands be tested as prediction-coding pairs. An alternative is to compute a fixed band ordering for a class of data, and use that fixed ordering when coding all data sets in that class. For instance, a fixed "CZCS ordering" could be pre-computed and used for all CZCS data sets. The ordering computed in this way from our sample data was given in Section 6. This has the benefit of being extremely fast (similar in speed to simply ignoring inter-band correlation); furthermore, our experiments have shown that for CZCS data, the results are almost indistinguishable from the optimal ordering results. Thus, this last approach of using a fixed band reordering seems to be a very promising lossless multispectral compression method.

8 Acknowledgments

The author would like to thank Doreen Revis for her help with various aspects of the CM-5, Jim Tilton, Manohar Mareboyana, Gene Feldman, and Mary James for supplying test data, and Sarah Blanton for providing helpful comments on the presentation of these results.

References

[1] B. R. Epstein, R. Hingorani, J. M. Shapiro, and M. Czigler, "Multispectral KLT-wavelet data compression for landsat thematic mapper images," in *Proc. of the Data Compression Conference (DCC '92)*, pp. 200–208, 1992.

[2] S. Gupta and A. Gersho, "Feature predictive vector quantization of multispectral images," *IEEE Trans. Geoscience and Remote Sensing*, vol. 30, May 1992.

[3] T. Markas and J. Reif, "Multispectral image compression algorithms," in *Proc. of the Data Compression Conference (DCC '93)*, pp. 391–400, 1993.

[4] R. E. Roger and M. C. Cavenor, "Lossless compression of imaging spectometer data," tech. rep., Department of Electrical Engineering, Australian Defence Force Academy, Canberra AACT 2600, Australia, 1993. This report may be obtained by anonymous FTP from `evans.ee.adfa.oz.au` in directory `pub/reports/remsen`.

[5] R. E. Roger and J. F. Arnold, "Reversible image compression bounded by noise," *IEEE Trans. Geoscience and Remote Sensing*, vol. GE-32, pp. 19–24, Jan. 1994.

[6] G. K. Wallace, "The JPEG still compression standard," *Communications ACM*, vol. 34, pp. 30–44, Apr. 1991.

[7] R. E. Tarjan, "Finding optimum branchings," *Networks*, vol. 7, pp. 25–35, 1977.

[8] P. M. Camerini, L. Fratta, and F. Maffioli, "A note on finding optimum branchings," *Networks*, vol. 9, pp. 309–312, 1979.

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman and Company, 1979.

[10] H. N. Gabow, "An efficient implementation of Edmond's algorithm for maximum matching on graphs," *J. ACM*, vol. 23, no. 2, pp. 221–234, 1976.

[11] H. N. Gabow and R. E. Tarjan, "Faster scaling algorithms for general graph-matching problems," *J. ACM*, vol. 38, no. 4, pp. 815–853, 1991.

[12] P. G. Howard and J. S. Vitter, "New methods for lossless image compression using arithmetic coding," *Information Processing and Management*, vol. 26, pp. 765–779, Nov. 1992.

[13] J. Makhoul, "Linear prediction: A tutorial review," *Proc. of the IEEE*, vol. 63, pp. 561–580, Apr. 1975.

[14] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text Compression.* Englewood Cliffs, NJ: Prentice Hall, 1990.

[15] R. F. Rice, "Some practical universal noiseless coding techniques," Tech. Rep. Publication 79–22, Jet Propulsion Laboratory, Mar. 1979.

[16] R. G. Gallager and D. C. V. Voorhis, "Optimal source codes for geometrically distributed integer alphabets," *IEEE Trans. on Info. Theory*, vol. IT–21, pp. 228–230, Mar. 1975.

[17] S. Todd, G. Langdon, and J. Rissanen, "Parameter reduction and context selection for compression of gray-scale images," *IBM J. Research and Development*, vol. 29, pp. 188–193, Mar. 1985.