Capturing Global Redundancy to Improve Compression of Large Images

Barbara L. Kess* Stephen E. Reichenbach
bkess@cse.unl.edu reich@cse.unl.edu
Dept. of Computer Science & Engineering
University of Nebraska – Lincoln
Lincoln, NE 68588-0115

Abstract

A Source Specific Model for Global Earth Data (SSM-GED) is a lossless compression method for large images that captures global redundancy in the data and achieves a significant improvement over CALIC and DCXT-BT/CARP, two leading lossless compression schemes. The Global Land 1-Km Advanced Very High Resolution Radiometer (AVHRR) data, which contains 662 Megabytes (MB) per band, is an example of a large data set that requires decompression of regions of the data. For this reason, SSM-GED compresses the AVHRR data as a collection of subwindows. This approach defines the statistical parameters for the model prior to compression. Unlike universal models that assume no *a priori* knowledge of the data, SSM-GED captures global redundancy that exists among all of the subwindows of data. The overlap in parameters among subwindows of data enables SSM-GED to improve the compression rate by increasing the number of parameters and maintaining a small model cost for each subwindow of data.

1 Introduction

Lossless compression of large data sets is an interesting research problem that affords the possibility for efficient compression using a source specific paradigm rather than the common universal modeling paradigm that is used for small images. The Global Land 1-Km AVHRR [1] data is a good example of a large data set. This is a multiband, remotely sensed data set in which each band contains 694,417,757 samples. The current plan is to produce, store, and make available for distribution a data set for

*Currently with Microsoft Corporation in Redmond, WA. Support was provided in part by NASA Graduate Student Researchers Program grant NGT-51293, a NASA Nebraska Space Grant Consortium fellowship, and the Center for Communication and Information Sciences (CCIS) at the University of Nebraska.

1068-0314/97$10.00 © 1997 IEEE

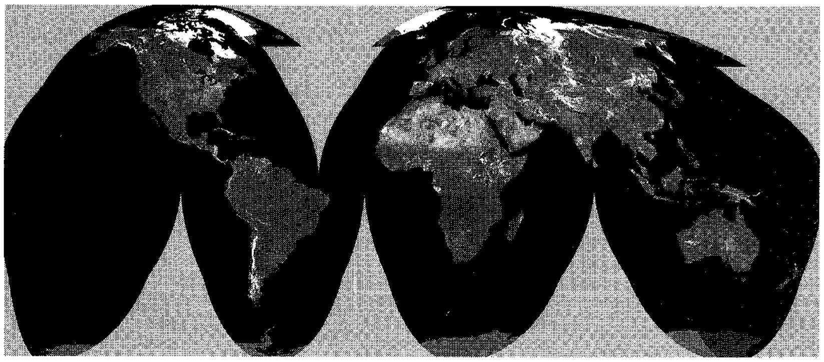


Figure 1: NDVI band of the Global Land 1-Km AVHRR data

10 day periods throughout the year. Because users want to view or analyze regions of the data, it is necessary that the compression scheme allows for decompression of subwindows of the data. This is accomplished by dividing each data set into a collection of subwindows and compressing each subwindow individually. The Normalized Vegetation Differential Index (NDVI) band for June 1-10, 1992 is shown in Figure 1.

Leading lossless compression schemes, that are designed to work well on a variety of small images, are based on the paradigm of universal modeling and coding. Rissanen and Langdon [9] define a universal compression scheme as one which compresses each data set with no *a priori* knowledge of the actual parameters in the data and is provably asymptotically optimal for data produced by an information source that fits the assumptions of the model. Some of the leading lossless compression methods are "good" schemes based on the principles of universal modeling, but have not been proven to be asymptotically optimal. Such schemes are referred to in [11] as intuitive modeling schemes. DCXT-BT/CARP [11] and CALIC [14] are two examples of "good", but non-optimal models that assume no *a priori* knowledge of the data. DCXT-BT/CARP is an adaptation of theoretical results on universal modeling [9, 10], to gray scale images. CALIC is a context-based scheme that groups contexts into equivalence classes and uses them in conjunction with a gradient-adjusted predictor and error-feedback.

Given enough data, a universal method will achieve an optimal coding rate. However, if the same data is compressed as a collection of subwindows, the compression rate for universal and intuitive methods is non-optimal if the size of the subwindow is not large enough for the model to learn the optimal parameters. In contrast to universal modeling, SSM-GED makes one pass through the large data set and creates the model parameters before compression of the subwindows. Given a subwindow size of 65,536 samples, SSM-GED achieves a better compression rate than CALIC and DCXT-BT/CARP. SSM-GED improves compression by increasing the number of parameters. Because there is global redundancy in the data, there is overlap in the model parameters among subwindows. Also, because one model is used for all of the subwindows, the model cost is shared over all of the subwindows. This allows an increase in the number of parameters for the model which results in a compression rate that is better than leading intuitive lossless compression methods.

This paper first describes the data. This is followed by an explanation of SSM-GED. Compression results are given for the Normalized Differential Vegetation Index (NDVI) band of the April and May 1992 data sets. These results are compared with CALIC and DCXT-BT/CARP.

2 The Global Land 1-Km AVHRR data

The Global Land 1-Km Advanced Very High Resolution Radiometer (AVHRR) data is an example of a large data set that is being collected, stored, and disseminated on a regular basis. This is a multiple-band, remotely sensed composite data set, produced for 10-day periods throughout the year, beginning with 1992 and continuing indefinitely. The data is currently stored in the Interrupted Goode's Homolosine map projection [2] and each data set is nearly 10 gigabytes (GB).

Each data set covers the entire globe and thus the changes in the data set are temporal in nature and not geographical. The image size is 17,347 lines by 40,031 samples. The data set contains five AVHRR spectral bands stored as 10-bit unsigned integers in two byte format and five additional bands (NDVI, satellite zenith, solar zenith, relative azimuth, and date) stored in unsigned byte format.

The number of bytes needed to store a band of this data set in unsigned byte format is 662 megabytes (MB). Each band stored in unsigned two-byte format requires 1,324 MB. The 10 bands of the data set contain 9.05 GB of data. The current plan is to produce, store, and make available for distribution a composite data set for 10-day periods throughout the year on a continuous basis, making compression of the data a practical necessity.

These data have the potential to give high compression ratios. The two-byte integer format for the 10-bit spectral bands do not utilize six bits per sample. There are also three fill or mask conditions in these data. Because these data are for the study of the world's land surface, all non-land regions are masked and set to a constant. In addition, the interrupted areas in the Interrupted Goode's Homolosine map projection are set to another constant. The third fill condition is over land where there are no data, such as in parts of Antarctica. The fill comprises approximately 80.1% of the image area.

Because of the size of this data set, users often want to extract subwindows of the data. For this reason, the image is compressed so that portions of the image can be decompressed efficiently. For example, if a user wishes to study a portion of South America, he is able to enter latitude/longitude coordinates and extract a subwindow of data that contains the area between the coordinates. It is highly impractical to decompress all of the data that precede the requested geographical area. Since the common coding algorithms, Huffman [3] and arithmetic [13], do not allow for random access to compressed data, one solution to the subwindow decompression problem is to divide the image into many subwindows or blocks of data, and compress each block independently. Provided the blocks are not too small, it is simple and practical to store a table of offsets for each block of data and retrieve only the necessary blocks during decompression.

Kess, Steinwand, and Reichenbach [5] give a compression method called Land & Mask for the Global Land 1-Km AVHRR data in which the data is compressed as a collection of subwindows. Their compression scheme is uses a hierarchical JPEG-based method [7] for compression of the land data and a quadtree for compression of the mask data. SSM-GED gives a 14.9 percent improvement over Land & Mask for non-hierarchical compression.

3 Model structure for SSM-GED

SSM-GED is a static, context-based tree model created prior to compression of subwindows of the data. Initially, SSM-GED makes one pass through all of the data, collecting conditional probability functions for all of the contexts of size four that actually occur in the data. SSM-GED then consolidates the tree from the bottom up by reducing the size of contexts for which the model is overfitted. The reduction procedure is based on estimates of model cost versus the entropy of the probability functions.

If the data fits a Markov process of order k, then for each sample x_n

$$P(x_n \mid x_{n-1},\ldots,x_{n-k}) = P(x_n \mid x_{n-1},\ldots,x_{n-k},\ldots,x_0). \tag{1}$$

A context tree for a Markov process, M_k, of order k on an alphabet S with $d = \mid S \mid$ symbols $x \in S$, is a d-ary tree of depth k in which each leaf node represents one of the d^k possible contexts $z = z_0,\ldots,z_{k-1}$. Each leaf node, n, contains a conditional probability function for the *next* symbols that occur after the context at node n.

Because the data is available prior to compression it is possible for SSM-GED to empirically create a context tree that represents a Markov process of order k, and then consolidate the tree from the bottom up to a more general tree structure. The number of contexts is a combinatorial explosion as the order increases. However, only a small subset of the possible contexts actually occurs, making it possible to empirically construct the tree, creating contexts as they occur.

Adaptive context algorithms such as DCXT-BT/CARP [11] and Algorithm Context [9] decide dynamically how large each context is based on previously seen data. Other algorithms such as CALIC [14] use a predetermined context size, but group the contexts into equivalence classes because the amount of data available for compression only allows enough data to learn the parameters for a small number of contexts.

Nohre [6] gives an optimal consolidation algorithm for a binary tree. However, for gray scale images in which the alphabet size is 256, the number of possible consolidated trees is a combinatorial explosion. Weinberger, Rissanen, and Arps [11] approach this problem by transforming the 256-ary gray scale tree into a binary tree. Although this allows for the use of binary tree consolidation methods, the original assumption that a binary tree is a good fit for gray scale data may not be correct. SSM-GED uses a consolidation algorithm for the context tree that achieves a compression rate that is close to the computed fourth order conditional entropy of the data, which is a lower bound for any compression scheme that uses the same four context samples [4] to model the data.

There are two phases for creating the model. The first phase is data collection and the second phase is context consolidation.

3.1 Data Collection

In the data collection phase there are three decisions. The first is how many nearby samples to include in each context, the second is which nearby samples to include, and the third decision is how to order the bits that occur in the context.

3.1.1 Determining the Context Size

The decision of how many samples to initially include in each context is decided empirically, based on available computing resources and the number of unique contexts that actually occur in the data. Contexts were collected for the NDVI band from six different data sets taken from the months of April and May 1992. Empirically, it was determined that collecting and manipulating contexts of size four is convenient with only a few gigabytes of disk space available. Table 1 shows that for the amount of data tested, the average number of parameters per context of size four is only 5.47. Thus, with a context size of four, many of the contexts have a limited number of samples in the context. Based on the rate of decrease in the number of parameters per context, the average number of parameters for contexts of size five will be between two and three.

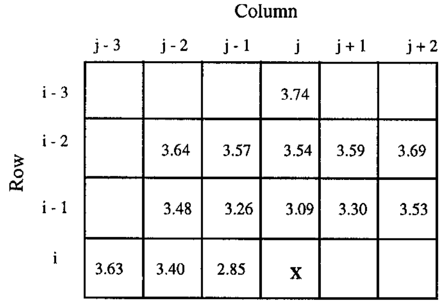
After the tree consolidation is finished, another pass is made through the data to collect three more context samples for the contexts of size four in the consolidated tree. These additional contexts are consolidated to context lengths of four through seven. Although a small estimated compression gain is observed by adding more contexts, SSM-GED uses a maximum context size of four in order to reduce the complexity of the model.

3.1.2 Choosing Samples for the context

The second decision is which four samples to collect for each context. Given x at row i and column j in the NDVI image I, Figure 2 shows the conditional entropy of sample x based on each of the causal samples, y, whose distance from x is less than or equal to three. These entropy results were obtained from the land samples in the April 1-10, 1992 NDVI band. The conditional entropies were gathered only on sample points in which all fourteen surrounding sample values were land values.

April through May 1992 NDVI Data, Band 6				
Context Size	Total Samples	Total Contexts	Conditional Entropy (Bits/Sample)	Average Parameters Per Context
4	4,166,058,204	8,298,971	0.4735	5.47
3	4,166,058,204	260,138	0.5120	13.81
2	4,166,058,204	15,987	0.5212	31.86
1	4,166,058,204	182	0.5964	83.88

Table 1: Conditional entropy based on empirical calculations



Row \ Column	j - 3	j - 2	j - 1	j	j + 1	j + 2
i - 3				3.74		
i - 2		3.64	3.57	3.54	3.59	3.69
i - 1		3.48	3.26	3.09	3.30	3.53
i	3.63	3.40	2.85	X		

Figure 2: Conditional entropy of x given y at neighboring locations

The conditional entropy given in Figure 2 is a measure of the amount of information that each sample contributes to x. The results show a slight bias in the horizontal direction for the NDVI data. This is a result of the instrument used in the image acquisition process. The results given in Figure 2 suggest that on the average, the best four context samples to choose with respect to each x at location (i, j) are the samples at locations $(i, j-1)$, $(i-1, j)$, $(i-1, j-1)$, and $(i-1, j+1)$.

For each $x = I(i, j)$, its context $Z(x)$ is initially chosen as

$$Z(x) = z_0, z_1, z_2, z_3 = I(i, j-1), I(i-1, j), I(i-1, j-1), I(i-1, j+1). \qquad (2)$$

3.2 Choosing a bit ordering

To determine a good bit ordering for the context samples, four different orderings of the bits were tested. It was decided from the empirical results that the best ordering uses the first two samples followed by the four high order bits of context samples two and three and the four lower order bits of context samples two and three. The reasoning for this approach is that the samples in locations $(i, j-1)$ and $(i-1, j)$ give the most information about the value of x respectively. The conditional entropy of x with respect to these samples is considerably less than samples z_2 and z_3. Samples z_2 and z_3 are close in the amount of information they contain about sample x. SSM-GED reorders the bits in $Z = z_0, z_1, z_2, z_3$ where $z_i = z_{i,0} \ldots z_{i,7}$ to form the context $Z^*(x)$ as follows:

$$Z^*(x) = z_{0,0} \ldots z_{0,7}, z_{1,0} \ldots z_{1,7}, z_{2,0} \ldots z_{2,3} z_{3,0} \ldots z_{3,3}, z_{2,4} \ldots z_{2,7} z_{3,4} \ldots z_{3,7}. \qquad (3)$$

3.3 Consolidating the Contexts

SSM-GED consolidates the context tree by examing each context to determine if it is most efficient to keep the context or make it smaller by removing the last byte of the context and moving its frequency counts to the parent context. Given a Markov tree M_k, of depth k, the algorithm begins by traversing through the parent nodes at depth $k-1$. Before examining each child node, n_i, of a parent node, n, the algorithm computes a probability function for the parent node by adding together all of the child probability functions. Each probability function consists of frequency counts for the values that appear after the context.

The decision to keep or remove the child node, n_i, is based on computing the model cost of the child node plus the number of bits required to code all of the samples counted in its probability function. The code length plus model cost estimate is compared to the number of bits required to code all of the samples at n_i using the probability function at the parent node.

For each context, the model cost is computed by precisely stepping through the model compression algorithm and counting the bits that will be used to store the context. The model cost is an estimate because a small part of the cost depends upon the final consolidated tree structure. The model compression algorithm is cogent with Rissanen's mean model cost of $0.5m \log n$ in [8] where m is the number of parameters and n is the number of samples. However, the model cost for each node is considered individually according to the model compression algorithm. This gives an estimate of the model cost for each context that is more precise than the mean model cost.

There are two aspects to compressing the model: compressing the structure of the tree and compressing each probability function that occurs in the tree. The tree compression algorithm is initialized with the maximum depth of the tree. It compresses the nodes in the tree in the order they occur during a breadth first search that starts with the root of the tree.

Some of the nodes have a probability function associated with the node. A table of distributions is created and the distributions are stored in the table in the same order that they appear during the breadth first search of the tree. When the model is decompressed, each node that is associated with a distribution will point to the correct distribution.

<table>
<tr><th colspan="7">April through May 1992 NDVI Data, Band 6</th></tr>
<tr><th>Context Size</th><th>Samples</th><th>Number of Contexts</th><th>Cond. Entropy (Bits/ Sample)</th><th>Average Parameters Per Ctxt</th><th>Model Cost Estimate (Bytes)</th><th>Total Compression Estimate (Bytes)</th></tr>
<tr><td>4</td><td>3,817,595,987</td><td>21,759</td><td>0.2148</td><td>31.56</td><td>1,285,178</td><td>103,790,863</td></tr>
<tr><td>3</td><td>238,600,804</td><td>9,477</td><td>3.5132</td><td>49.08</td><td>793,769</td><td>105,575,437</td></tr>
<tr><td>2</td><td>109,268,538</td><td>6,826</td><td>4.0089</td><td>59.34</td><td>664,192</td><td>55,419,750</td></tr>
<tr><td>1</td><td>590,057</td><td>147</td><td>4.7508</td><td>88.18</td><td>19,260</td><td>369,662</td></tr>
<tr><td>0</td><td>2,818</td><td>1</td><td>6.2740</td><td>137.00</td><td>148</td><td>2,358</td></tr>
<tr><td>Total</td><td>4,166,058,204</td><td>38,210</td><td>0.503873</td><td></td><td>2,762,547</td><td>265,158,070</td></tr>
</table>

Table 2: Coding and model cost estimates after consolidation

4 Results

Table 2 gives the results of the consolidation algorithm. These results tell how many samples and contexts are represented at each level of the tree along with an estimate of the model cost in bytes. The model cost is estimated at 2,762,547 bytes. The total actual model cost for SSM-GED is 2,788,021 bytes which includes 2,744,220 bytes plus 43,801 bytes for distributions of border samples. The border sample distributions were not included in the original model cost estimate. The model cost estimates are very close to the actual model cost.

Table 3 gives the results from compression of the data as a collection of subwindows with 256 rows and 256 columns. The actual compression results are 1.2 MB higher than the compression estimate. This increase is because the border samples in each subwindow are coded at a slightly higher bit rate than estimated by the consolidation process. The bit rate of the border samples was not considered in the compression estimates, making the model applicable to other subwindow sizes.

Table 3 compares the compression results to those obtained with DCXT-BT/CARP and CALIC using the same subwindows of data. The results show an overall 7.26% improvement over CALIC and a 10.92% improvement over DCXT-BT/CARP. Although results are not given, CALIC performs better on the subwindows than it does on the entire image without subwindows.

SSM-GED has an efficient execution time, making it a practical algorithm to use for compression and decompression of data. It's memory requirement is proportional to the number of parameters used in the model.

5 Summary

Compression and decompression of the Global Land 1-Km AVHRR data presents a practical example of the need to develop a new paradigm for compression of large

Compression Results for Global Land 1-Km AVHRR Data
NDVI Band, April through May 1992
Data Size: 6 bands with 694,417,757 bytes per band
Results include 42,712 bytes for indexing the subwindows
Model cost for SSM-GED is 2,788,021 bytes

NDVI Data	DCXT- BT/CARP	Bit Rate	CALIC	Bit Rate	SSM-GED	Bit Rate
April 1-10, 1992	46,055,687	0.5306	45,420,578	0.5233	41,619,611	0.4795
April 11-20, 1992	45,432,072	0.5234	44,916,353	0.5175	41,097,508	0.4735
April 21-30, 1992	47,062,516	0.5422	46,487,665	0.5356	42,698,423	0.4919
May 1-10, 1992	48,887,319	0.5632	48,198,931	0.5553	44,277,990	0.5101
May 11-20, 1992	51,890,359	0.5978	50,153,501	0.5778	45,711,438	0.5266
May 21-30, 1992	59,674,604	0.6875	52,032,792	0.5994	48,167,085	0.5549
Total	299,002,557	0.5741	287,209,820	0.5515	263,572,055	0.5061
Total With Model					266,360,076	0.5114

Table 3: Compression results using subwindows with 256 rows and 256 columns

images. It is a large data set in which a large amount of data is available prior to compression and small subwindows of the data set are decompressed independently. A universal modeling scheme would compress each individual subwindow of data without regard to structure that exists in the larger global set of data and in succeeding data sets. SSM-GED achieves a better compression rate than leading universally based models because it preprocesses the data and creates statistical parameters that are used during compression and decompression of the individual subwindows of data. SSM-GED is able to improve the compression rate by increasing the number of parameters, but still maintaining a small model cost per subwindow of data.

The results have potential application to other large data sets, such as video, in which there is a large sampling of the data available prior to compressing individual frames of the data. Witten, Moffat, and Bell apply the concept of a static model plus individually decodable units to large document databases in [12].

References

[1] Jeff C. Eidenshink and John L. Faundeen. The 1-Km AVHRR Global Land Data Set: First stages in implementation. *International Journal of Remote Sensing*, 15(17):3443–3462, 1994.

[2] J. P. Goode. The Homolosine projection: A new device for portraying the Earth's surface entire. *Association of American Geographers, Annals*, 15:119–125, 1925.

[3] D. A. Huffman. A method for the construction of minimum redundancy codes. *Proceedings IRE*, 40:1098–1101, 1952.

[4] Barbara L. Kess. *A Source-Specific Model for Compression of Global Earth Data.* PhD thesis, University of Nebraska at Lincoln, May 1997.

[5] B.L. Kess, D.R. Steinwand, and S.E. Reichenbach. Compression of the Global Land 1-Km AVHRR Dataset. *International Journal of Remote Sensing*, 17(15), October 1996.

[6] Ragnar Nohre. *Some Topics in Descriptive Complexity.* PhD thesis, Linkoping University, 1994.

[7] William B. Pennebaker and Joan L. Mitchell. *JPEG Still Image Data Compression Standard.* Van Nostrand Reinhold, 1993.

[8] J. J. Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information Theory*, IT-30(4):629–636, July 1984.

[9] J. J. Rissanen and Glen G. Langdon, Jr. Universal modeling and coding. *IEEE Transactions on Information Theory*, IT-27(1):12–23, January 1981.

[10] Marcelo J. Weinberger. A universal finite memory source. *IEEE Transactions on Information Theory*, 41(3), May 1995.

[11] Marcelo J. Weinberger, J. J. Rissanen, and Ronald B. Arps. Applications of universal context modeling to lossless compression of gray-scale images. *IEEE Transactions on Image Processing*, 5(4):575–586, April 1996.

[12] I.H. Witten, A. Moffat, and T.C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images.* Van Nostrand Reinhold, 1994.

[13] I.H. Witten, R.M. Neal, and J.G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, June 1987.

[14] Xiaolin Wu and Nasir Memon. CALIC - a context-based, adaptive, lossless image coding scheme. *Accepted for publication in the IEEE Transactions on Communications*, May. 1996.