

Compression of Regions in the Global Advanced Very High Resolution Radiometer 1-Km Data Set

Barbara L. Kess¹
University of Nebraska - Lincoln
Lincoln, Nebraska 68588-0115
bkess@cse.unl.edu

Daniel R. Steinwand²
EROS Data Center
Sioux Fall, South Dakota 57198
stein@suno.cr.usgs.gov

Stephen E. Reichenbach
University of Nebraska - Lincoln
Lincoln, Nebraska 68588-0115
reich@ser.unl.edu

ABSTRACT

The global advanced very high resolution radiometer (AVHRR) 1-km data set is a 10-band image produced at USGS' EROS Data Center for the study of the world's land surfaces. The image contains masked regions for non-land areas which are identical in each band but vary between data sets. They comprise over 75 percent of this 9.7 gigabyte image. A quad tree is used to find and compress boundaries for land and masked regions. The mask is compressed once and stored separately from the land data which is compressed for each of the 10 bands. The mask is stored in a hierarchical format for multi-resolution decompression of geographic subwindows of the image. The land for each band is compressed by modifying the method described in Kess, Steinwand and Reichenbach (1994) to ignore fill values. This multi-spectral region compression efficiently compresses the region data and precludes fill values from interfering with land compression statistics. Results show that the masked regions in a one-byte test image (6.5 Gigabytes) compress to .2 percent of the 557,756,146 bytes they occupy in the original image, resulting in a compression ratio of 89.9 percent for the entire image.

1. INTRODUCTION

The Global Advanced Very High Resolution Radiometer (AVHRR) 1-km project is an example of the need for data compression in the Earth Observing System Distributed Information System (EOSDIS). As part of this project, the U.S. Geological Survey's (USGS) Earth Resources Observation Systems (EROS) Data Center, in conjunction with other international data centers and science groups, is planning to produce global data sets at 1-km resolution, one data set per 10-day period. This data set contains just less than 10 gigabytes of data. Without any compression, the data set requires at least 15 CD-ROMs that hold 660 MB each. The requirements for compression of this data set include lossless decompression of geographic subwindows of the data at multiple resolutions. Compression methods that divide the image into blocks and compress each block with a hierarchical format that allows multiresolution decompression have been developed (Kess, Steinwand, and Reichenbach, 1994).

Since the purpose of this data set is for study of the world's land surfaces, all non-land regions are masked and set to a constant. Mask values are used to fill regions of water, unused parts of the framed data in the map projection, and land where there is no data. The masked regions are exactly the same in all 10 bands of the image, but may vary between data sets. They

¹ Hughes STX Corporation. Work performed under consultant agreement no. 93-9002-11904.

² Hughes STX Corporation. Work performed under U.S. Geological Survey contract 1434-92-C-40004.

comprise at least 75 percent of the image, making efficient compression of the fill areas a major factor in the success of the compression algorithm. A quad tree is used to describe and compress boundaries for the masked regions and land regions. Since the masked regions are exactly the same in all ten bands of the image, it is only necessary to compress the region data once for the entire image, rather than once for each band. The mask data is stored separately from the land data and is accessible during decompression of each of the 10 bands. The following describes the approach used for separating the region data from the land data. Results are given for a 10-band test image containing one-byte data in all 10 bands, at 6.5 Gigabytes. The full 9.7 Gigabyte image (which contains 5 bands with two-byte integer pixels) was not yet available for our test purposes.

2. COMPRESSION OF REGIONS

The quad tree used to compress the region boundaries is a region quad tree as described in Samet (1984). This 4-way tree structure represents a recursive decomposition of the image into quadrants. When each of the four child quadrants of a parent node are found to be homogeneous, the parent node is used to represent the information present in all four child quadrants. In the following example, solid regions are shown with black nodes and non-solid regions are shown with white nodes. Black nodes that are close to the root of the tree represent large solid regions of data (See fig. 1).

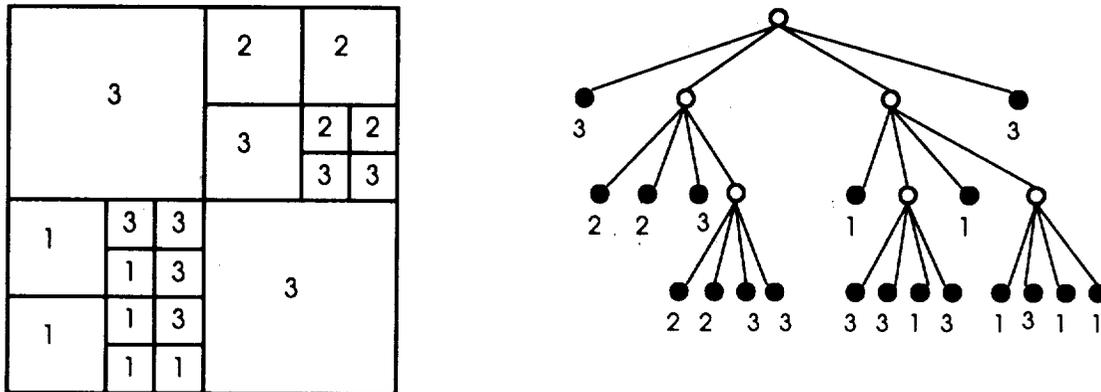


Figure 1. Regions of an 8x8 block and its respective quad tree.

The levels of the quad tree can be easily used to store data for resolution levels that differ by a factor of four. Each internal node stores a subsampled value from its four children. The subsampling method chooses the upper left pixel in each 2 x 2 block, which means that each internal node in the quad tree receives the value of the first child node.

The mask compression algorithm initializes each leaf node with the value of the pixel it represents. All leaf nodes that represent land pixels receive a constant that represents land regions. Thus, each leaf node is marked as being part of one of the four possible regions: water, land, land with no data, and unused parts in the framed map projection.

The tree is built from the leaf nodes up to the root, giving each parent the value of its first child and setting each internal node's solid flag to true if all four children are solid and have the same value. Blocks of size $2^n \times 2^n$ require $2^{2n} + 2^{2n}/3$ nodes to build the tree. The testing was done with a block size of 128×128 , which requires 21,845 nodes. This makes it feasible to build and store the tree in memory during compression and decompression (See fig. 2).

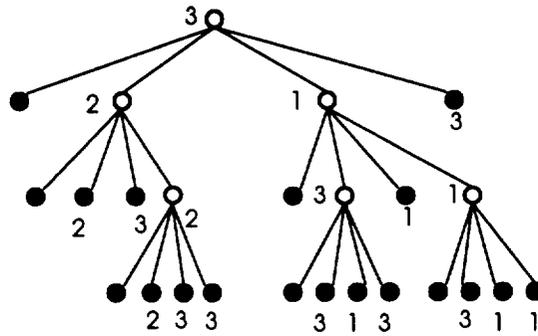


Figure 2: Quad tree from Figure 1 with value in first child promoted to parent mode.

To compress the tree a breadth first search is done, starting at the root. Each node sends a maximum of 3 bits to the output stream. The first bit specifies whether the node is solid (or not solid) and two more bits are used to give the value represented by the node. A queue is used to determine the visiting order for the nodes. If a node is not solid, it enqueues each of its children. If a node is solid, none of its children are enqueued because all necessary information for reconstructing its children has already been given to the output stream. If a node is a first child it does not send its value to the output because its parent's value has already been compressed. If a node is a leaf, it does not send a solid bit because it has no children and only represents one value. The worst scenario with this tree is that there are no solid regions in which case two bits are transmitted for each sample in the original block of data, plus 1 bit to designate that each internal node is not solid. If n is the number of samples in the original image then the maximum number of bits used for the compressed data is $2n + n/3$. The compressed bit stream is shown in Figure 3. Each row represents the bits used to compress a level of the tree which was shown in Figure 2.

```

011
1 010 001 111
1 110 111 010 1 011 101 001
10 11 11 11 01 11 11 01 01

```

Figure 3: Compressed bit stream for quad tree in Figure 2.

3. LAND COMPRESSION

The land compression algorithm compresses each block with a hierarchical method proposed by Sloan and Tanimoto (1979). The pixels are reordered by placing pixels needed for the coarsest resolution at the beginning of the block, followed by pixels needed to fill in the next resolution, until the full resolution image is restored losslessly. The block is then de-correlated with a JPEG prediction scheme that predicts each pixel based on the value of the previous pixel (Wallace, 1993). The decorrelated data is coded with Huffman coding (Huffman, 1962).

Since the region data is already compressed, the land compression algorithm needs only to compress land data. Some blocks, however, contain a mixture of land data and fill data. In these cases, the land compression algorithm is modified to ignore the fill data. During de-correlation and coding, each pixel is tested to determine if it is a land value or a mask value. All mask values are ignored during decorrelation and coding, producing compressed data that contains only land values. This improves the compression statistics for each band because no

extra space is given to fill data and the presence of fill data does not affect statistics used to compress the land data.

4. MASK DECOMPRESSION

Decompression of each block compressed with the mask separation approach involves first decompressing the mask and then filling in the land values where they belong. Prior to decompression of specific blocks, the quad tree is created in precisely the same manner as during compression. If the user specifies a resolution other than 1 km for the decompressed image, then the number of leaf nodes is computed to match the number of pixels in the decompressed image. Since all blocks are decompressed to the same resolution, the quad tree has the correct size for each block. The algorithm finishes when it has traversed all of the leaf nodes, so it automatically decompresses each block to the correct resolution. Each leaf receives the offset into the decompressed block where its pixel value belongs. During compression, pixels were transferred from the image to the leaf nodes of the tree. Now, during decompression, the pixel values are transferred from the leaf nodes to their appropriate offset in the decompressed block.

Once the tree is created, the compressed information is read and used to fill in the nodes of the tree. Every internal node enqueues its four children into the queue to be visited. Each node, except for the root node, checks the solid flag in its parent. If its parent is solid, then it simply inherits the parents solid flag and node value. If the parent node is not solid, then the child node receives bits from the compressed data, 1 bit for the solid flag and 2 bits for the value. Some exceptions to this are that the first child inherits the value from its parent rather than retrieving it from the compressed data, and leaf nodes do not retrieve a solid bit. After all nodes of the quad tree have been visited, the leaf values are ready to be copied to the correct offset in the decompressed block. A constant value is placed into each pixel that requires a land value.

5. LAND DECOMPRESSION

After the mask decompression routine has stored region data in the decompressed block, the land decompression routine decompresses the land values and places them into pixels that contain a constant value, representing land. For each land value decoded from the compressed input, the algorithm computes its offset into the decompressed block. If this position contains a mask value, the algorithm moves to the next position. Decompressed samples are only allowed to be copied to pixels that contain a land constant.

6. BLOCKS OF UNEVEN SIZE

The global image dimensions are not evenly divisible by 2^n . The compression algorithm is designed for blocks whose dimensions are $2^n \times 2^n$. This leaves blocks on the right and bottom sides of the image that are not full. The mask compression and decompression algorithm accommodates these blocks by adding pad values to the tree. If a leaf value falls into the padded area it is noted as such and ignored when the values are assigned to internal nodes of the quad tree. This maintains the integrity of the quad tree, but does not send any bits to the compressed output for the padded pixels. When the tree is recreated during decompression it knows exactly which leaf nodes fall into the padded areas and ignores them when copying leaf values to the decompressed block.

The land compression and decompression algorithm does not depend on a $2^n \times 2^n$ block size. Changing the block size does not affect the land algorithm's ability to reorder the data during compression and to put the data back into the correct position during decompression.

7. RESULTS

Data from a 10-band image with one byte data in each band was compressed with the hybrid approach described in Kess, Steinwand, and Reichenbach (1994) and also with the mask separation approach described in this paper. The hybrid approach compresses blocks that contain only two or three distinct values with run length encoding and it compresses solid blocks with two bytes. The other blocks are compressed with the land compression algorithm. The header bytes are used to store the block table and a global Huffman table. The number of bytes out is the actual space required for the data in the image.

Compression with Hybrid Approach

<u>Band</u>	<u>Bytes In</u>	<u>Bytes Out</u>	<u>Header Bytes</u>
1	694,417,757	100,498,557	172,352
2	694,417,757	108,417,401	172,352
3	694,417,757	91,753,214	172,352
4	694,417,757	94,351,654	172,352
5	694,417,757	94,554,288	172,352
6	694,417,757	83,834,431	172,352
7	694,417,757	71,682,592	172,352
8	694,417,757	56,498,487	172,352
9	694,417,757	56,875,854	172,352
10	694,417,757	58,676,725	172,352
TOTAL	6,944,177,570	817,143,203	1,723,520
Total Compressed Size:		818,866,723 bytes	
		780.932 megabytes	
Compression Ratio:		88.21%	

The mask separation approach uses the approach described in this paper in which region data is compressed separately from the land data. In the results using the mask separation approach, the number of bytes out for each band is the amount of space required to compress the land data using a JPEG prediction scheme and global Huffman coding. Preliminary results using an adaptive Huffman algorithm (not reported here) instead of global Huffman coding indicate at least a 20 megabyte improvement for the entire image.

Compression with Mask Separation Approach

<u>Band</u>	<u>Bytes In</u>	<u>Bytes Out</u>	<u>Header Bytes</u>
Mask	694,417,757	992,345	170,304
1	694,417,757	88,893,680	171,328
2	694,417,757	96,178,846	171,328
3	694,417,757	77,461,785	171,328

4	694,417,757	78,718,869	171,328
5	694,417,757	79,199,998	171,328
6	694,417,757	68,360,373	171,328
7	694,417,757	65,899,562	171,328
8	694,417,757	45,091,220	171,328
9	694,417,757	49,054,680	171,328
10	694,417,757	47,687,610	171,328
TOTAL	6,944,177,570	697,538,968	1,883,584
Total Compressed Size:		699,422,522 bytes	
		667.021 Megabytes	
Compression Ratio:		89.93%	
Improvement of Mask Separation to Hybrid Approach:		14.59%	

Distribution of Pixels in Each Band

<u>Pixel Type</u>	<u>Bytes</u>	<u>% of Image</u>
Unused portions	177,245,765	25.52%
Water	368,222,266	53.03%
Land without data	12,288,115	1.77%
Land	136,661,611	19.68%
Total	694,417,757	100.00%
Total Mask Bytes:	557,756,146	
Mask Compression Ratio:	99.82%	

8. REFERENCES

- D. A. Huffman, 1962, "A Method for the Construction of Minimum Redundancy Codes", *Proceedings IRE*, v. 40, 1962, pp. 1098-1101.
- B. L. Kess, D. R. Steinwand, and S. E. Reichenbach, 1994, "Compression of the Global Advanced Very High Resolution Radiometer 1-Km Data Set", in *International Geoscience and Remote Sensing Symposium Proceedings*, August 1994.
- W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.
- H. Samet, "The Quadtree and Related Hierarchical Data Structures", *Computing Surveys*, v. 16, no. 2, June 1984.
- K. T. Sloan and S. L. Tanimoto, "Progressive Refinement of Raster Images", *IEEE Transactions on Computers*, November 1979, v. C-28, no. 11, pp. 871-4.