

**MARS 2001 ODYSSEY  
GAMMA COMPRESSION ALGORITHM**

11/01/2002

Prepared by:  
Mars Observer Team

The data compression algorithm utilized by the Gamma Ray Remote Sensing Spectrometer (GRS) instrument on the Mars Observer mission, called Y9, was created by R.F. Rice of the Jet Propulsion Laboratory. It is one example from a class of data compression algorithms referred to as universal noiseless coders in that they work equally well on all data sets without regard to the signal's underlying entropy and are error-free, i.e., the signal is 100% restorable. This algorithm compresses the GRS data by removing nearly all of the redundancy in the data spectrum so that only the actual information content (or entropy) in the spectrum is telemetered.

Y9 is an adaptive data compression algorithm. As configured for GRS, it is really a collection of simple, small coders which when selected properly, adapt to the wide entropy range expected to be exhibited by the GRS data. For another data set which does not exhibit the 0.1 to 8-bits per sample entropy range of GRS, the data compression algorithm would not have to be as internally adaptive nor include so many coder options. R.F. Rice and JPL have demonstrated this by using a much simpler version of Y on the Voyager mission.

Basically, Y9 is a code substitution compressor. It estimates what symbols are most frequently represented in the data stream and substitutes a shorter symbol for them. In other words, if the 8-bit symbol a appears in the raw GRS gamma spectrum 1200 times, substituting a 3-bit symbol b for a will save

$$1200 * 5 = 6000 \text{ bits}$$

This is an oversimplification of the algorithm but it does go far to explain the core of the compression scheme. Much of the rest of the algorithm is involved with choosing the optimum substitutions because as alluded to above, the same compression scheme is not used over the entire 16,000 channel GRS raw data spectrum.

There are no user adjustable parameters for regulating the compression algorithm. The algorithm is almost devoid of parameters and the few that are available were shown to have almost no effect on the compression of GRS data. The adaptability of the algorithm enables it to compress raw GRS gamma spectra which exhibit wide variations in entropy from spectrum to spectrum and even within a single spectrum.

The one parameter which most strongly influences the algorithm's compression efficiency is the number of different symbols in the spectrum. Because Y9 is a code substitution algorithm, if the number of symbols in the spectrum goes up, the frequency of each symbol decreases so the benefits of substituting shorter symbols is lost.

For example, if there are just four symbols in the entire GRS gamma spectrum, then no more than two bits per symbol are necessary to convey all of the information in the spectrum. Table B-1 presents an example of a data stream with 94 symbols in it. In the first column are the four possible symbols,  $a$ . Here the symbols are represented by 2-bit binary numbers. The second column is the number of times each symbol appears in the data stream. Column three shows a symbol,  $b$ , which could be substituted for the original symbol. Finally, the last column tallies the number of bits in the hypothetical compressed spectrum.

Table B-1  
Compressing A Four Symbol Data Stream

raw symbol, $\alpha_n$	F frequency of $\alpha_n$	Substituted symbol, $\beta_n$	B bits in $\beta_n$	F * B
00000000	71	1	1	71
00000001	18	01	2	36
00000010	4	001	3	12
00000011	1	000	3	3
	94			122

In this example, the maximum number of bits to transmit the 94 symbols in the spectrum is  $94 * 2 = 188$  bits if the unnecessary and redundant, 6 most significant bits (left-most) of each  $a$  is truncated. But substituting the symbols shown compresses the spectrum to just 122 bits.

The decompression algorithm can distinguish the  $\beta_n$  symbols by scanning the compressed spectrum from most significant to least significant bit (left to right) until it identifies a unique symbol. It may have to scan more than one bit before it can uniquely distinguish the symbol from all other ones. The symbol set(s) used must be known to both the compression and decompression functions. This is accomplished by choosing between simple symbol sets built into the compression algorithm. Unlike some other data compression schemes, it is not necessary to downlink the symbol table along with the compressed data.

Now if the number of symbols present in the data stream is increased to eight, the size of the compressed spectrum may grow to 291 bits if symbols are substituted as shown in Table B-2.

Table B-2  
Compressing An Eight Symbol Data Stream

raw symbol, $\alpha$	F Frequency of $\alpha$	Substituted symbol, $\beta$	B bits in $\beta$	F * B
00000000	23	1	1	23
00000001	20	00	2	40
00000010	16	010	3	48
00000011	13	0110	4	52
00000100	10	01110	5	50
00000101	6	011110	6	36
00000110	4	0111110	7	28
00000111	2	0111111	7	14
	94			291

If the 5 most significant bits of each  $\alpha$  are truncated, then the raw spectrum could be transmitted in  $3 * 94 = 282$  bits. In fact, Y9 might perform such a compression and downlink the 282-bit compressed spectrum. This simple bit truncation compression is one of the alternatives available within Y9 to compress a gamma spectrum. So, even in a higher entropy case such as in the second example, Y9 eliminates redundancy from the data.

The symbols used for the substitution in these examples are not the output of the algorithm. They are used only to illustrate that the length of the compressed spectrum grows as the number of unique symbols in the raw spectrum increases. As the number of symbols needed to convey the information in the data stream grows, the entropy, or amount of information, in the data stream is increasing. It is loosely analogous to adding precision to an engineering measurement which in turn increases the number of significant figures.