Binary variable order adaptive algorithms like UMC (Rissanen 1986) and JBIG can be used to losslessly compress non-binary data by splitting the data into planes, each of 1 bit resolution, and passing each plane to a separate instance of the algorithm. The UMC algorithm operated in this way is the most powerful lossless signal data compressor the authors are aware of. In this paper, we attempt to develop an understanding of why this approach is so effective. We investigate the common technique of Gray coding the data before splitting it into single-bit planes and passing to the modeler and coder, and compare it to a simple weighted binary coding. We then propose a non-binary pseudo-Gray code as a method of generating planes of resolution greater than or equal to 1 bit, and compare it with the other conventional methods.

Splitting an r-bit input into r planes results in r separate streams that are each easier to code in that each histogram more quickly attains values that well represent the distribution of the binary data in the plane. However, the sum of the codelengths may still not be less. Because we may be disregarding relationships between the planes, the resulting probability distribution (after scaling and summing the individual random variables) may not be the same as the distribution of the r-bit data. A histogram of the full resolution data approximates this distribution more closely, but takes much longer to properly train.

The above idea motivates the consideration of a middle path that uses a non-binary splitting of the r-bit input. Our algorithm to generate the pseudo-Gray code is much the same as that for the construction of a binary Gray code, except that instead of minimizing the Hamming distance between neighboring bit planes, we instead minimize the Euclidean distance between adjacent groups of bit planes. Specifically, the algorithm is as follows:

1. Start with the all zero code word corresponding to the source zero.
2. Form the next codeword by changing the least significant group by the least amount that results in an unused code word; the change can be either an increment (add 1) or a decrement (subtract 1) of the group value; if a change does not generate a new code word, move to the next higher significance group.

It was found that some combinations of planes with resolution higher than 1 bit resulted in better compression than the binary Gray code method when coding the following 8-bit grayscale images: pentagon (512×512), lena (256×256), photographer (256×256), barb (512×512), mandrill (512×512), and a texture pattern (512×512). The performance improvement ranged from 1.6% (barb) to 6.7% (texture), but was typically above 3.5%.