Error-Resilient Block Sorting

Lee Butterman
Stuyvesant High School
New York, NY 10282-1099
lbutte01@utopia.poly.edu

Nasir Memon
Polytechnic University
Brooklyn, NY 11201
memon@poly.edu

“Blocksorting Transform” (BWT) is a lossless compression technique that has become increasingly popular in recent years due to the fact that it achieves high performance compression (almost as good as in PPM, with fast speed (not much slower than gZip). It is perhaps the first lossless compression algorithm to combine these two properties. BWT breaks its input into large blocks, and performs an invertible transform on each block. The modified blocks are more compressible with fast and simple means, like Move-To-Front (MTF) coding, and/or arithmetic coding.

Although BWT provides high compression at low computational cost, it suffers from a problem common to most general purpose lossless compression techniques. If the Inverse BWT detects an error, it must discard the entire block and lose up to almost a megabyte of data (the common blocksize for large files). The amount of data lost can be reduced by reducing the block size, but this results in a price in terms of compression performance. In fact experiments reported in the literature have shown the compression performance of BWT drops off rapidly with decreasing block size.

In this paper we propose error-resilience techniques for BWT that allow for final losses to be roughly proportional to the channel losses. (We assume throughout that the size of the compressed file does not change, but only the contents; we assume that there are only size-preserving errors.) Our modification to BWT (called δελ), involves the addition of error-resilience information that allow graceful degradation of error corrupted BWT compressed files. In fact, we show that by adding about 3 to 9% additional information in BWT, we can recover 90% of the original file even when losing a significantly large number of bytes from the compressed file. This overhead does not include that needed to recover from errors in the entropy coded stream. There are many techniques given in the literature for recovering from errors in Huffman coded or arithmetic coded streams and the proposed techniques can be applied in conjunction with any one of them. For the sake of completeness, however, we do provide experimental results that include a simple variable to fixed length arithmetic coding scheme that provides error resiliency in the entropy coded stream.

The full version of this paper can be found at www.stuy.edu/~butterm1/erbs.pdf