Multistream Compression[1]

Jiří Kochánek[a], Jan Lánský[b], Petr Uzel[b], Michal Žemlička[b]

[a]UniControls, a.s., Praha, Křenická 2257, 100 00 Praha 10, Czech Republic
[b]Charles University, Faculty of Mathematics and Physics,
Malostranské nám., 118 00 Praha 1, Czech Republic
kochanek@unicontrols.cz, zizelevak@gmail.com,
petr.uzel@centrum.cz, michal.zemlicka@mff.cuni.cz

We present a data transformation [1] and compression method based on decomposition of input stream into multiple streams of counters of passes through individual nodes of the specific binary tree. The streams are coded separately – using methods that can compress them best. The use of multiple streams has given the method its name: multistream compression (MC). The method differs from other models based on splitting data into different streams by the fact that in this case the streams do not contain symbols, but counters.

Compression is quite complex and requires multiple passes. First the file is scanned to get frequencies and positions of first occurrences of individual symbols. The frequencies and positions are used to build binary tree. In the second pass the tree is traversed from root to leaf representing the read symbol. During the traversal the counters in each visited node are increased and some of them are written into node-specific streams. These streams are then analyzed and a best-suited method for their compression (like Elias codes and other integer encodings) is selected. In the third pass the counters from the streams are coded, merged, and written into the final output stream by the way which allows efficient streamable decompression

Decompression is much simpler: the input contains information necessary for reconstruction of the binary tree used for stream separation/collection and data of the streams merged in proper order to generate the original message. The decompression is therefore single pass and requires less computing power and less space than the compression does.

We have tested the method alone and in combination with other transformations BWT, MTF, and RLE. As a test set we have used files from Silesia corpus.

All the combinations started with BWT and MTF. Then RLE and Huffman coding or RLE and arithmetic coding, or MC were used. The last combination of transformations was the most successful: in overall by 0.2 or 0.18 bpc. Also in compression of individual files the last combination had always the best compression ratio of the three mentioned combinations.

The other domain is the compression of AC and DC parts of signal in JPEG images. According our simple tests it is possible to compress them better using the new method than by Huffman or arithmetic coding as it is common. Using MC in codecs (instead Huffman or arithmetic coding) is seems hopeful.

References

[1] Jiří Kochánek. Způsob transformace a bezztrátové komprimace dat v elektronické podobě (in Czech). Patent application PV 2007-114; Czech Office of Industrial Ownership, 2007.

[1] This research was partially supported by Charles University Grant Agency in the project "Text compression" (GAUK no. 1607, section A) and by the Program "Information Society" under projects 1ET100300419 and 1ET100300517.

1068-0314/08 $25.00 © 2008 IEEE
DOI 10.1109/DCC.2008.21